commodore

COMPUTER CLUB

20 L. 3.000

La rivista degli utenti di sistemi Commodore

Mensile - maggio 1985 - Anno IV - N. 20 - Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe

Come costruire un videogame

Gestione archivi su disco

Le finestre nello schermo

Nuovi programmi per la scuola

5 giochi originali per Vic, C16 e C64

In edicola questo mese

commodore
COMPUTER CLUB

PRESENTA

62 Programmi

per il

VIC 20
C 16
PLUS 4

Sommario

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Giovanni Bellù, Andrea e Alberto Boriani, Giancarlo Castagna, Eugenio Coppari, Marco De Martino, Luca Galluzzi, Giancarlo Mariani, Flavio Molinari, Enrico Scelsa, D. Matturro, M.L. Nitti, Fabio Sorgato, Danilo Toma
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Impaginazione/Illustrazioni: Francesco Amatori, Renato Caruso
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Mirco Croce (coordinatore), Giuseppe Porzani, Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio - Segretaria: Lilliana Degiorgi
● Roma: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Abbonamenti: Marina Vantini
Tariffe: prezzo per copia L. 3.000. Abbonamento annuo (11 fascicoli) L. 28.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 55.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl
Fotolito: Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/1/82 - Direttore Responsabile: Michele Di Pisa
Distribuzione: MePe: via G. Carcano 32 - Milano Sped. in abb. post. gr. III - Pubblicità inferiore al 70%

LE IMMAGINI DI QUESTO FASCICOLO

BESSIE TRA GUERRA E PACE

Un ragazzo e il suo personal computer alle prese per sbaglio (e per gioco) con l'elaboratore centrale del pentagono. Uno scienziato impazzito (il progettista dell'elaboratore) e le sue teorie sui giochi.

Si chiamava Wargames: forse il film più suggestivo uscito finora sul calcolatore, le sue possibilità, i suoi possibili errori. Evidentemente gli sceneggiatori si erano studiati a fondo la storia dell'elaborazione dei dati. Perchè dopo il censimento americano di inzio 900 (vi ricordate, ci eravamo fermati lì, la volta scorsa) la storia del computer intreccia sviluppo tecnologico a strategie di guerra, diffusione velocissima di nuove macchine a ricerche affascinanti di grandi, geniali e bizzarri scienziati.

Dalle macchine a schede perforate ai primi calcolatori elettromeccanici, il salto è breve. E' il 1936. Konrad Zuse, 26 anni, tedesco, sequestra il salotto dei suoi genitori e inizia a trafficare con cavi e relè. La sua creazione si chiama Z1, e combina per la prima volta controllo programmato e rappresentazione binaria dei numeri. Konrad non lo sa, ma nella stessa direzione lavorano in quel momento scienziati di nazionalità diverse, e non certo nel salotto di casa.

Nel 1943 J. Presper Eckert, John W. Mauchly e Herman H. Goldstine dell'Università della Pennsylvania propongono all'esercito degli Stati Uniti il progetto di una macchina capace di risolvere ad altissima velocità i problemi balistici d'artiglieria. Si chiama ENIAC, ed è il primo calcolatore elettronico della storia. In un secondo effettua 300 moltiplicazioni, pesa oltre 30 tonnellate, impiega 18 mila tubi elettronici e occupa 180 metri quadrati di superficie. Per dimensiononi e capacità, l'ENIAC rappresenta un progresso immenso rispetto ad esempio al Mark 1, calcolatore elettromeccanico realizzato dall'Università di Harvard e dalla Ibm nel 1944.

Una tappa fondamentale viene toccata dal genio matematico dello scienziato ungherese John Von Neumann, che realizza EDVAC: per la prima volta un elaboratore memorizzava non solo i dati su cui lavorare ma anche le istruzioni per il suo funzionamento. Un'idea che ispira anche gli elaboratori successivi.

Mac, il robot californiano, ama le bellezze al bagno ma non può avventurarsi troppo verso la linea dell'acqua. E' la didascalia alla foto apparsa nel 1949 su Scienza Illustrata. Già allora le macchine automatiche scatenavano l'immaginazione. In particolare il robot, parola che deriva dal cecoslovacco "robota". Significato: schiavitù, lavoro forzato

Un elaboratore di bordo e una rete di computer a terra consentono all'astronauta americano Edward White la sua mitica passeggiata nello spazio, nel 1965. Rispetto al progetto Mercury, la tecnologia ha già fatto passi da gigante. Un esempio sono le foto scattate nello spazio, che vengono registrate in forma numerica su nastro e trasmesse a terra via radio. Un elaboratore provvede poi a riportare i numeri sotto forma di immagine. Le tonalità possibili tra bianco e nero sono 64, ogni foto è composta da 40 mila punti. La luna è vicina.

Quelli della prima generazione (agli inzi degli anni Cinquanta). Quelli della seconda, che costituiscono la base del progetto spaziale Mercury. Quelli della terza generazione, nata con il lancio del sistema Ibm 360 (1964) che impiegano circuiti microminiaturizzati.

E infine, le idee accumulate in 300 anni di storia si sono sintetizzate nei computer di quarta generazione, quelli che si trovano sui nostri tavoli.

AVVERTENZE IMPORTANTI

Alcuni lettori richiedono, mediante la scheda dell'ultima pagina, argomenti che sono trattati proprio nel fascicolo da cui è tratta la stessa scheda. A volte, infatti, l'argomento richiesto viene ampiamente illustrato come vero e proprio articolo oppure mediante risposta ad un lettore.

Si prega, pertanto, allo scopo di evitare inutili ripetizioni, di leggere attentamente queste pagine perchè, nella scelta delle domande che selezioniamo per la pubblicazione, privilegiamo, ovviamente, quelle comuni alla maggior parte dei lettori.

Sprite

□ **Vorrei sapere che cosa è esattamente uno sprite e se limita la versatilità del mio C-16 non possedere sprite.** *(Francesco Iozia - Ispica)*

• Uno sprite è una "figura" programmabile che, con semplici istruzioni può essere spostata in lungo e in largo sullo schermo del televisore. Il principale vantaggio di uno sprite consiste nel fatto che se viene fatto spostare a destra non è necessario provvedere a "cancellare" la sua immagine dalla posizione precedente.

Se, ad esempio, vuoi spostare un carattere di venti "posti" lungo una riga dello schermo, puoi ricorrere ad un programmino del genere:

```
100 PRINT CHR$(147)
110 FOR I+1 TO 20
120 PRINT "A";130 FOR K=1 TO 50
140 NEXT K
150 NEXT I
```

La prima riga cancella lo schermo, mentre la 110, con la 150, provvede a stampare uno dopo l'altro i venti caratteri "A" grazie alla presenza del carattere di punto e virgola (;) di riga 120. Le righe 130 e 140 generano un "ritardo" in modo da rendere visibile l'animazione. In questo modo, però, il carattere "A" si muove verso destra ma lascia la sua presenza dietro di sè.

Per ovviare a questo inconveniente aggiungi le seguenti linee:

```
125 PRINT CHR$(157)
145 PRINT CHR$(32)
```

facendo attenzione a digitare i caratteri di punto e virgola.

Il carattere CHR$(157) sposta il cursore alla sua sinistra mentre CHR$(32) rappresenta lo spazio vuoto (come quello prodotto dalla barra spaziatrice).

Il risultato che si ottiene è l'animazione del carattere "A" che sembra spostarsi verso destra mentre, invece, è la successione di scritture e successive cancellazioni in diverse celle video del televisore.

E' chiaro che le cose si complicano notevolmente se si desiderano effettuare spostamenti in alto, in basso o in diagonale. Figurarsi, poi, il caso in cui si desiderano spostare più caratteri contemporaneamente allo scopo di generare immagini più grandi!

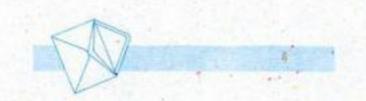
I vantaggi degli sprite che, presenti solo nel Commodore 64, possono essere ben otto contemporaneamente, evitano tecniche complicate di programmazione e consentono, tra l'altro, di sovrapporsi, senza cancellarli, ai caratteri alfanumerici presenti sullo schermo e di farli riapparire quando si allontanano.

Se ciò non bastasse, tieni presente che gli sprite si possono muovere di un pixel alla volta e non di otto come nel programmino prima esaminato. Un carattere, come è noto, può occupare una ben definita cella di memoria a differenza dello sprite che può sovrapporsi ad uno qualunque dei 64000 puntini elementari dello schermo del C-64.

Il C-16 non possiede gli sprite ma non per questo devi considerarlo limitato. Molti giochi possono esser creati mediante la ridefinizione dei caratteri come si può fare con il Vic 20 che vanta giochi non privi di interesse.

Imparare il Basic

□ **Sono un ragazzo di 12 anni e possiedo un C-16 ma da mesi cerco disperatamente qualche cassetta magnetica che insegni il linguaggio Basic. Dove trovarle, se esistono?** *(Massimiliano Bonizzoni - Pontalbera)*

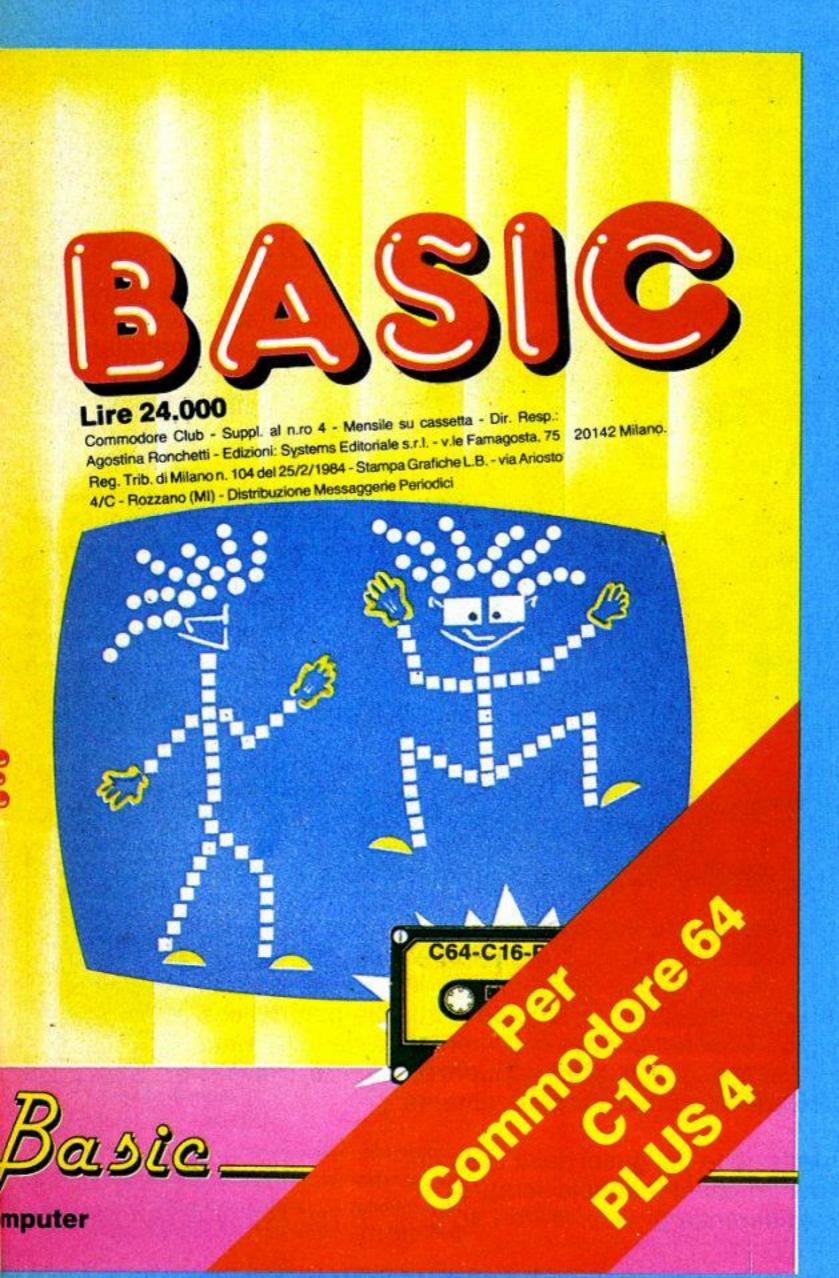
• Per venire incontro alle esigenze dei principianti (e non) come te, è in distribuzione nelle edicole un corso dal titolo "24 ORE BASIC" interamente su nastro. E' sufficiente caricare in successione le varie cassette della confezione per comprendere l'uso delle istruzioni del tuo computer. Non dimenticare, però, di tenerti aggiornato seguendo le pagine di questa rivista...

Per i nuovi commodoriani

Possessori dei nuovi nati da Mamma Commodore (il C 16 ed il Plus 4) avranno sicuramente notato che accanto al pulsante d'accensione della macchina si trova un pulsantino bianco. E' il famigerato tasto RESET che consente di ripristinare tutti i vari puntatori utilizzati dal Computer durante il suo lavoro. In pratica premendo il pulsante RESET è come se si spegnesse e riaccendesse il Calcolatore.

In realtà non accade proprio questo in quanto la RAM utente non viene a perdere l'alimentazione elettrica di cui ha bisogno per vivere e quindi non viene inizializata. Quello che voglio dire è che se vi era in memoria un programma questo vi rimane, anche se per poterlo nuovamente utilizzare si deve ricorrere ad alcuni trucchi (o ad appositi programmi

IN EDICOLA

TUTTO I
IN 4

L BASIC
NASTRI

che implementino la funzione OLD).

Ora, avendo, sia il C 16 che il Plus 4, su ROM interna un programma monitor per il linguaggio macchina, ho scoperto un piccolo trucchetto che consente di recuperare un programma che abbia fatto andare in "Crash" la macchina. E' infati sufficiente premere il tasto di <RUN/STOP> posto sulla sinistra della tastiera e contemporaneamente il pulsante RESET.

Questo ci permette di entrare in Monitor e quindi, se ne usciamo con il tasto <x> seguito da <RETURN>, e chiediamo il LIST vedremo che il nostro programma è lì bello e pronto per essere salvato e/o modificato.

Ancora sull'auto-repeat

☐ **Sul N. 14 a proposito dell'auto-repeat, scrivete che si può ottenere col comando POKE 650,128. So, però, che è possibile ottenerlo anche con POKE 650,255 ed annullarlo con POKE 250,32. Chi ha ragione?** *(R. Ventura - Siracusa)*

• Tutti e due. Della locazione 650, infatti, è sufficiente settare (mettere al valore unitario) il settimo bit per abilitare la funzione. Ciò vuol dire che qualsiasi numero superiore a 127 è adatto allo scopo come pure un qualsiasi valore inferiore a 128 disabilita l'auto repeat per tutti i tasti. Altri valori limitano l'auto-repeat alla pressione della barra spaziatrice, ai tasti di controllo cursore eccetera.

Morsetti del registratore

☐ **Che tensioni sono presenti sui morsetti del registratore e che cosa avviene esattamente quando si preme uno dei suoi tasti?**

• Sul manuale del C-64 (e del Vic 20) è riportato lo schema del connettore per il registratore. Si tenga presente che il circuito stampato, benchè sia a doppia faccia, presenta connessioni tra i pin superiori ed inferiore (sono, cioè in "corto"). Utilizzare la parte superiore od inferiore è, di conseguenza, la stessa cosa.

Ciò premesso precisiamo che:

1) la massa è comune (GDN piedini A-1);

2) la tensione di +5 volt presente sui pin B-2 è utilizzata, insieme con la massa, per alimentare i circuiti integrati del registratore.;

3) il motore viene invece alimentato, quando serve, dai pin C-3;

4) sui pin D-4 ed E-5 sono presenti, a seconda dei momenti, i segnali di invio o lettura dati;

5) i pin F-6, se collegati... brutalmente con la massa GND da un opportuno deviatore, "comunicano" al computer che un tasto del registratore è stato premuto. Se infatti apri il registratore puoi renderti conto, utilizzando un comune tester, di quanto asserito.

Stai attento, comunque, ad intervenire sui circuiti o sul connettore: un banale corto circuito, specie nella sezione alimentatrice, può mettere fuori uso il tuo personal!

Petspeed e comando Dim

☐ **Posseggo il compilatore Petspeed per il C-64 e segnala un errore nella seguente riga di un mio programma:**
10 M=t400:DIM D$(M)
Potreste aiutarmi dato che non riesco a capire bene il libretto di istruzioni in inglese? *(Giuseppe Greco - Paestum)*

• Non è possibile dimensionare "dinamicamente" un vettore (numerico o stringa) usando il Petspeed. Ciò vuol dire che l'argomento del comando DIM deve essere esplicito e non riferito ad una variabile. Nel tuo caso devi modificare la riga come segue:
10 M=400: DIM D$(400)

Non sempre, purtroppo, il valore dell'argomento è noto a priori, come nel caso da te proposto. In alcuni programmi di matematica, ad esempio, il valore dell'argomento è noto dopo alcune elaborazioni effettuate in una parte del programma stesso. In questi casi è bene studiare quale può essere il valore massimo, nel corso di un'elaborazione, e digitarlo tra parentesi. E' ovvio che è indispensabile fare i conti con la memoria disponibile altrimenti, se si esagera, si rischia un messaggio di OUT OF MEMORY ERROR.

Registrare l'azimut

☐ **Esiste un programma o un "qualcosa" per tarare l'azimut di un registratore?** *(Bruno Gandolfi - Nizza Monf.)*

• Il sistema c'è ma, nella sua semplicità, richiede purtroppo l'utilizzo di strumenti di misura che non tutti possiedono.

E' infatti sufficiente procurarsi un nastro su cui è inciso un (lungo) programma o file di dati registrati mediante un registratore perfettamente tarato (altrimenti l'operazione risulta del tutto inutile).

Collegando i morsetti di un oscilloscopio, e azionando il registratore in modo PLAY, è possibile, ruotando lentamente la vite di regolazione della testina, individuare la sua posizione ottimale in corrispondenza del massimo segnale che risulta dallo schermo dell'oscilloscopio.

Se hai un amico che ha l'hobby dell'elettronica, e che possiede l'oscilloscopio, ti potrà facilmente aiutare. E per gli altri? La System editoriale (cioè noi) metterà tra breve sul mercato...

Fuori gli autori!

In seguito ad alcune involontarie omissioni, di cui ci scusiamo con i lettori e con gli interessati, precisiamo che gli autori del volume GUIDA AL COMMODORE 64, (edizioni Mc Graw-Hill), recensito sul volume 15 di C.C.C. sono J. Heiborn, R. Talbott

L'autore della serie di articoli sul Kernal (N. 17, 18) è Giovanni Verrelli di Gioia Del Colle (BA) il cui nome (ed indirizzo) è già comparso sul N. 49 della rivista Micro & Personal Computer.

Le nuove Maxell

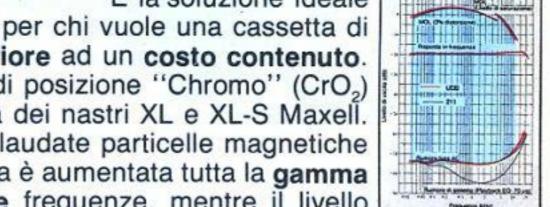
UD I

Vanta una tecnologia senza confronti nella sua categoria. Infatti tutte le proprietà magnetiche del suo nastro sono esaltate al massimo grazie alla nuova particella magnetica "Ferricrystal" che, a differenza di quelle tradizionali, è **assolutamente non porosa**. La UD I, quindi, offre un livello d'uscita migliorato su tutta la gamma di frequenze, specialmente nelle medie e basse; una più ampia gamma dinamica, ed ottime caratteristiche di lownoise. Adotta la meccanica P.A. (PHASE ACCURACY) per garantire la massima stabilità di svolgimento del nastro e per contenere la differenza di fase tra i canali stereo entro i 10°. Può essere usata su qualsiasi tipo di registratore.

UD II

È la soluzione ideale per chi vuole una cassetta di **categoria superiore** ad un **costo contenuto**. Utilizza un nastro di posizione "Chromo" (CrO_2) prodotto con la stessa tecnologia dei nastri XL e XL-S Maxell. Infatti il nastro della UD II è composto dalle collaudate particelle magnetiche "Fine Epitaxial", ulteriormente perfezionate. Di conseguenza è aumentata tutta la **gamma dinamica** del nastro, il livello d'uscita alle **medie** ed **alte** frequenze, mentre il livello di rumore di bias ed il rumore di modulazione sono ridotti praticamente a **zero**.

Il meccanismo di scorrimento P.A. e la perfezione dei gusci della cassetta assicurano un regolare svolgimento del nastro, mantenendolo sempre perpendicolare alla testina del registratore (differenza di fase entro i 10°).

maxell®

È TUTTA UN'ALTRA MUSICA.

TITOLI IN LINGUA ITALIANA

J. Heilborn, R. Talbott
GUIDA AL COMMODORE 64
pag. 440 L. 36.000
ISBN 887700001-5

R. Jeffries, G. Fisher, B. Sawyer
DIVERTIRSI GIOCANDO CON IL COMMODORE 64
pag. 280 L. 22.000
ISBN 887700004-X

H. Peckham
IL BASIC E IL COMMODORE 64 IN PRATICA
pag. 312 L. 27.000
ISBN 887700009-0

P. Hoffman, T. Nicoloff
IL MANUALE MS-DOS
pag. 264 L. 25.000
ISBN 887700018-X

NOVITÀ LIBRI

K. Skier
L'ASSEMBLER PER IL COMMODORE 64 E IL VIC-20
pag. 400 L. 35.000
ISBN 887700011-2

P. Scharf
GENITORI NELL'ERA DEL COMPUTER
pag. 208 L. 19.000
ISBN 887700023-6

NOVITÀ SOFTWARE

A. Bleasby
ASSEMBLER/DISASSEMBLER PER IL COMMODORE 64
L. 24.000
ISBN 887700904-7

distribuzione in libreria:
Messaggerie Libri S.p.A.
Via Giulio Carcano, 32
20141 MILANO MI
tel. 02 8438141-8467341, telex 310672 MESSIT I

McGRAW-HILL BOOK COMPANY GmbH
Lademannbogen 136
D-2000 Hamburg 63
REPUBBLICA FEDERALE TEDESCA
tel. +49 40 5382081, telex 2164048 MHBC D

Vettori

□ **Che cosa è un vettore?** *(Massimo Marcato Padova)*

• Nella sintassi Basic il vettore è un insieme di elementi che possono essere valori numerici o stringhe.

Esempio:

AB=34: XX=9.76: G2=-67.8
eccetera.

In questo modo risulta però piuttosto difficile rintracciare un dato tra cento soprattutto se si intende ricorrere a ricerche veloci che utilizzano cicli FOR...NEXT

Per ovviare a questo inconveniente il Basic consente di utilizzare il comando DIM che, compatibilmente con la memoria disponibile nel computer, permette di memorizzare un numero elevatissimo di elementi e di rintracciarli velocemente e facilmente.

Un esempio di utilizzo del comando DIM può essere il seguente:

```
100 DIM A(100)
110 PRINT "1-SCRIVI"
120 PRINT "2-LEGGI"
130 INPUT D
140 IF D=1 THEN 170
150 IF D=2 THEN 200
160 GOTO 110
170 INPUT "SCRIVO NUMERO";X
180 INPUT "IN POSIZIONE"; PS
190 A(PS)=X: GOTO 110
200 INPUT "LEGGO POS."; X
210 PRINT A(X): GOTO 110
```

La prima riga (100) riserva spazio nella memoria del computer. Le righe 110-160 rappresentano un mini-menu che "rifiuta" un qualsiasi valore diverso da "1" e da "2".

Se hai scelto di scrivere un numero (opzione "1" righe 110 e 140) il programma salta alla riga 170 ove chiede il numero (X) da memorizzare nella posizione PS del vettore (riga 180).

Nella riga 190 l'istruzione A(PS)=X va così interpretata: "Nella posizione PSesima del vettore A deve essere memorizzato il numero X"

E' possibile, ovviamente, leggere il numero presente in una qualsiasi posizione selezionando l'opzione 2 ed indicandola quando appare la domanda "LEGGO POS.?" (riga 200).

Vengono automaticamente emessi messaggi di errore se si tenta di dimensionare più di una volta lo stesso vettore, indicando un posizionamento maggiore di quello dimensionato (più di 100 nel caso specifico) oppure assegnando un valore negativo all'argomento racchiuso tra parentesi.

EDITORIALE · EDITORIALE

ORIALE · EDITORIALE

Giocando S'impara

Col proliferare di riviste di informatica è piuttosto improbabile che l'utente medio di un personal computer pretenda di trovare sulle riviste "su carta" ciò che può interessarlo sulle riviste su cassetta.

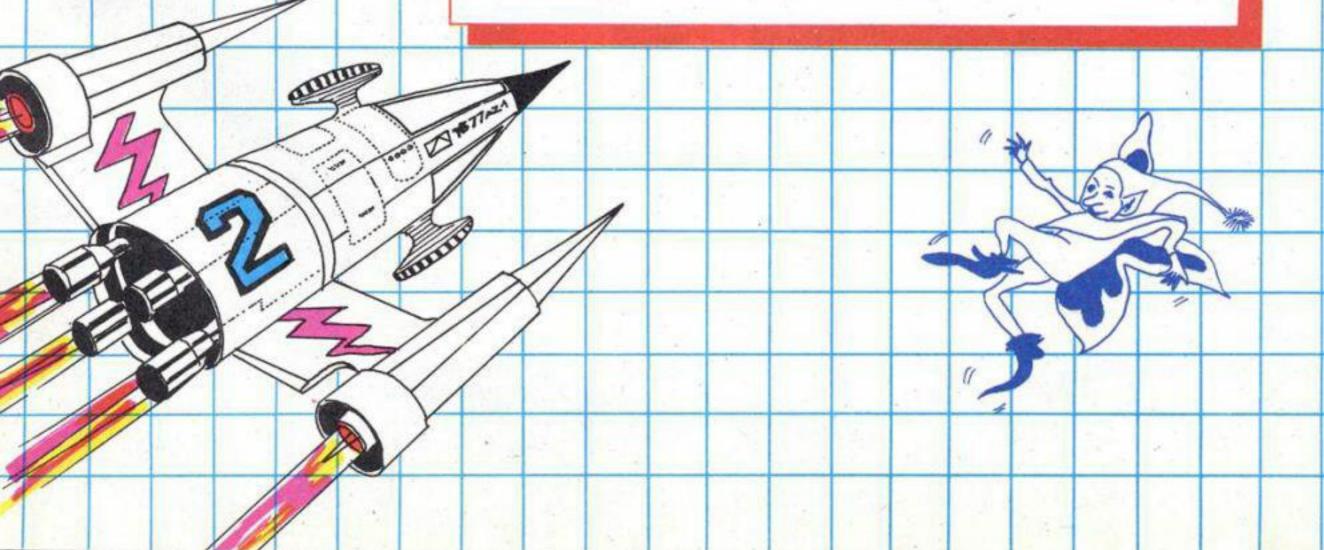
Queste ultime, come la nostra "Commodore Club", si rivolgono agli utenti che, volendo programmi complessi o sofisticati (e comunque troppo lunghi da digitare senza errori) individuano nel supporto magnetico il mezzo ideale per entrarne in possesso. Ne consegue che i giochi, i programmi e gli argomenti trattati su Commodore Computer Club non potranno mai competere con giochi tipo Honda o La voce (ci riferiamo ad alcuni notevoli giochi delle cassette di "Commodore Club") che richiederebbero la digitazione di diverse migliaia di codici in linguaggio macchina.

Viceversa, anche se decidessimo di riempire in tal modo una decina di pagine, il modesto risparmio di denaro non giustificherebbe la trascrizione del listato nè sarebbe, da parte nostra, possibile anche la pubblicazione del come funziona il gioco.

Questa rivista, pertanto, si rivolge ai possessori di apparecchi Commodore che, affamati di chiarezza e semplicità, desiderano sapere *come* creare programmi oppure il *perchè*, seguendo una certa procedura, si ottengono determinati risultati. In modo da sviluppare, nei *propri* programmi, idee in queste pagine affrontate in modo volutamente semplice e "povero".

E se i giochi pubblicati ti sembrano troppo brevi, privi di effetti sonori o altro, che aspetti a modificarli, personalizzandoli?

Non dimenticare che, secondo le moderne teorie psicologiche, il gioco, lungi dal costituire semplice svago, è utilissimo per acquistare esperienze altrimenti non conseguibili. In conclusione gioca pure quanto vuoi, ma con intelligenza: dopo un po' di tempo "rompi" il listato che hai appena digitato per vedere come è fatto dentro...

Alessandro de Simone

La rivista Commodore, consorella di Commodore Computer Club, ha promosso nei mesi passati una rubrica basata sui contributi provenienti dai propri lettori.

Questa sezione era destinata a tutti quei programmi, composti da un'unica riga, attinenti a qualsiasi argomento scelto dall'utente di un computer Commodore.

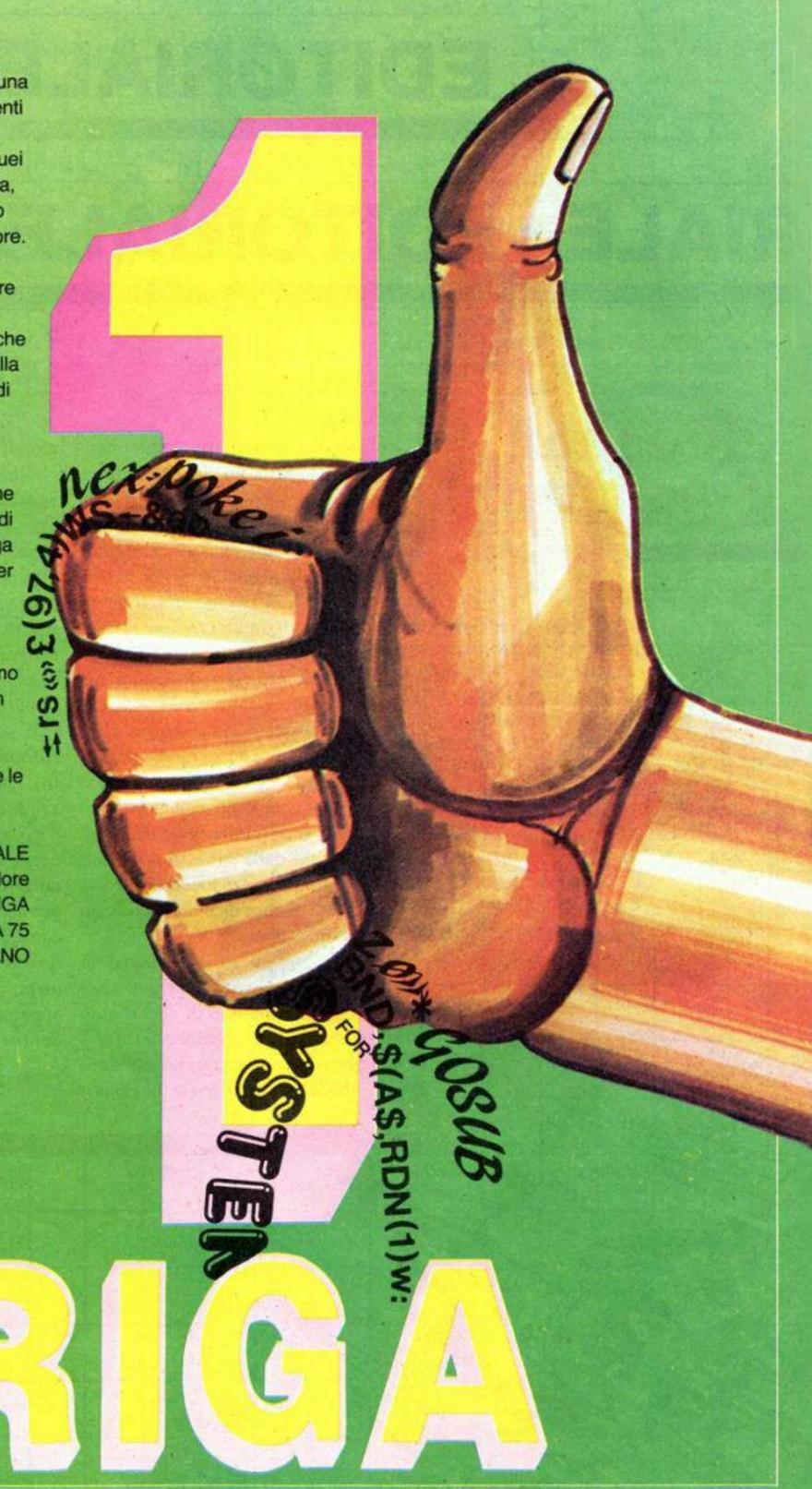
La novità e l'immediatezza di questa iniziativa, ci ha consentito di riscuotere un'enorme successo presso i lettori.

Le caratteristiche sono le medesime che aspirano il lavoro di tutti i redattori della rivista Commodore Computer Club, di cui Voi siete affezionati lettori.

Per questo motivo, nonch
e l'intercambiabilit
a che guida il lavoro di tutte le persone che collaborano con questa gamma di riviste, ci ha convinto a spostare 1 riga sulle pagine di Commodore Computer Club.

Speranzosi di incrementare ulteriormente i successi precedentemente riportati, Vi invitiamo a inviarci i vostri brevi programmi con l'unica limitazione che essi siano contenuti in un'unica riga di listato.

L'indirizzo a cui dovrete far pervenire le vostre "fatiche" è:

SYSTEMS EDITORIALE
Sezione Commodore
Rubrica 1 RIGA
V.le FAMAGOSTA 75
20142 MILANO

1

Catastrofe. li sconvolgenti effetti di un terremoto vengono ricreati tramite il vostro Commodore 64.

(La REDAZIONE)

```
1 FOR I=0 TO 255:PRINT
  I;:POKE 53270,I:NEXT
  :GOTO 1
```

Dedalo. Assisterete alla creazione di un'interessante labirinto.

(La REDAZIONE)

```
1 PRINTCHR$(205.5+RND(
  1));:GOTO 1
```

2

Caratteri casuali. Il vostro Commodore 64 crea caratteri casuali sul video.

(La REDAZIONE)

```
1 T=RND(1):POKE (T*100
  0+55296),RND(1)*16:P
  OKE (T*1000+1024),RN
  D(1)*254+1:GOTO 1
```

3

Titolatrice. Sul vostro schermo vedrete scorrere una fantastica scritta.

(La REDAZIONE)

```
1 FOR S=0 TO 6.4 STEP
  .3:PRINT TAB(10*COS(
  S)+16);"TITOLATRICE"
  :FOR T=0 TO 10:NEXTT
  :NEXTS:GOTO 1
```

4

Logaritmo. Calcolate il logaritmo di un numero nella base che più vi aggrada.

(Roberto Malamisura)

```
1 INPUT "LOG DI ";A:IN
  PUT "IN BASE";B:R=LO
  G(A)/LOG(B):PRINTR:G
  OTO 1
```

5

Inverti. Inserite una parola e i suoi caratteri verranno immediatamente invertiti tramite questo semplice listato.

(Roberto Malamisura)

```
1 INPUT A$:M=LEN(A$):D
  IM B$(M):FOR X=M TO
  1 STEP -1:B$(X)=MID$
  (A$,X,1):PRINTB$(X);
  :NEXT
```

6

Integrali. Ecco un'interessante programma che consente di calcolare gli integrali definiti.

Dovrete inserire gli estremi di integrazione (A e B).

(Michele Nannipieri)

```
1 DEF FNF(X)=SIN(X):L=
  (13-5)/100:FOR I=0 T
  O 99:X=I*L+L/2+A:Y=L
  *FNF(X):N=N+1:NEXT:P
  RINTN
```

7

Rotazioni. Al centro del vostro video comparirà una piacevole girandola.

(Luigi Dallanoce)

```
1 PRINT"[CLEAR]":A$="/
  -\|":FOR B=1 TO 10↑3
  8:FOR C=1 TO 4:PRINT
  "[CLEAR]"SPC(252)MID
  $(A$,C,1):NEXTC,B: R
  UN
```

8

Verme. Un lungo vermicello percorrerà tutto il video del vostro computer.

(La Redazione)

```
1 D$="[LEFT]":FOR X=1
  TO 200:PRINT"[CLEAR]
  ":PRINTD$;"O●O●O●O●O
  ";:D$=D$+"[RIGHT]":N
  EXT
```

9

Coordinate polari. Un interessante problema di analisi matematica: la conversione da coordinate Cartesiane in coordinate polari.

Vi verrà richiesto di inserire quest'ultime in INPUT.

(Giuseppe Randazzo)

```
1 INPUT X,Y:AL=ATN(Y/X
  )+(π/2)*(2- SGN(Y)*(
  1+ SGN(X))):PRINTSQR
  (X↑2+Y↑2),AL+180/π
```

10

Alfabeto. Questo breve programma vi consentirà di visualizzare, in progressione, tutte le lettere dell'alfabeto.

(Toresi Renzo)

```
1 FOR L=1 TO 200:FOR U
  =1024 TO 2023:POKE U
  ,L:NEXTU:FOR I=2023
  TO 1024 STEP -L:POKE
   I,32:NEXTI,L
```

11

Equazioni. Ecco un facile programma per calcolare le equazioni di secondo grado sul vostro Commodore 64.

(Giuseppe Mari)

```
1 PRINT"[CLEAR]AX↑2+BX
  +C[DOWN]":INPUT A,B,
  C:H=B↑2/4-A*C:K=ABS(
  H)↑.5:J=-B/2*A:PRINT
  J+K,J-K:IF H<0 THEN
  PRINT"I"
```

12

Barre. Sincronizzate il vostro televisore mediante questo semplice software.

(La Redazione)

```
1 POKE 646,N:FOR X=0 T
  O 39:PRINT"[RVS] ";:
  NEXT:N=N+1+16*(N=15)
  :GOTO 1
```

Teo Rusconi ha appena sfatato la leggenda secondo la quale i floppy disc sono tutti uguali

Difatti sembrano tutti uguali finchè non si osserva con attenzione il jacket. Qui termina l'uguaglianza.

La maggior parte delle società costruttrici sigillano i dischi un punto qui, un punto là, lasciando parte dei lembi non sigillati.

Prima o poi ai lembi accadono cose naturalissime: si gonfiano, si curvano, si raggrinziscono... in poche parole si aprono.

Con penne, matite, unghie persino un ragazzino di quattro anni come Teo può infilarsi in quegli spazi aperti.

Naturalmente è un danno enorme perchè se si inserisce qualcosa di molle e slabbrato nel disc-drive quest'ultimo può incepparsi; si può rovinare la testina e si possono perdere i dati.

GLI ALTRI DISCHETTI chiusi un punto qui, un punto là lasciano gran parte dei lembi aperti.

Questo può accadere con gli abituali sistemi di chiusura ma non con i dischetti Memorex che usa un procedimento esclusivo chiamato "Solid-Seam Bonding".

Con questo sistema ogni singolo millimetro quadrato dei lembi di tutti i dischi Memorex viene sigillato ermeticamente, rendendoli più rigidi e più resistenti.

DISCHETTI MEMOREX con lembi completamente saldati su tutta la superficie.

È un sistema che consente al floppy disc di sostenere ogni assalto, che impedisce alla testina di rovinarsi e ai dati di andare perduti.

Il che sta a dimostrare che un floppy disc Memorex non è uguale a tutti gli altri: è migliore. E il sistema di saldatura è solo un esempio della cura infinita con cui viene prodotto ogni floppy disc Memorex; sia esso da 8", da 5 1/4" o il nuovo 3 1/2".

Questa estrema accuratezza dà la garanzia che ogni disco Memorex è al 100% perfetto.

La prossima volta che acquistate un floppy disc - o qualche centinaio - ricordate: non tutti i dischetti sono uguali... Memorex vi mette al riparo da qualsiasi inconveniente.

A Burroughs Company

BURROUGHS-MEMOREX S.P.A.
Divisione Computer Media
Via Ciro Menotti, 14 Tel. 02/718551
20129 MILANO MI

COMPUTER QUESTO MESE È QUESTO

COMPUTER

N. 74 - lire 3500

il newsmagazine dell'informatica

I magnifici 7 1984

Sapresti riconoscerli?

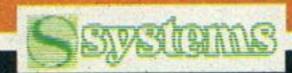
Quel fiscale d'un computer

Storia di Unix

Dischi laser

Il personal contabile

Benchmark: AlphaMicro AM-1000

Computer - Anno X - N. 74 - 25 marzo/25 aprile 1985 - Sped. Abb. Post. Gr. III/70 CH - Distr. MePe

Gogo the Ghost

Questo nuovo gioco prodotto dalla casa di software Firebird e distribuito dalla Mastertronic ci introduce nel mondo dell'occulto e del terrore.

Con il joystick o la tastiera dobbiamo infatti aiutare un piccolo fantasma a raggiungere la Principessa dei suoi sogni attraverso 150 stanze di un castello infestato da orribili creature. Sarà letale per GOGO, il nostro fantasma, entrare in contatto con questi mostri. Deve anche stare attento alle varie insidie che caratterizzano ogni stanza. Cancelli che si aprono e si chiudono, lance che attraversano l'intera stanza, palle d'acciaio. Per passare nelle stanze successive bisogna raggiungere la zona priva di pareti.

E' possibile rendersi invulnerabile agli oggetti diventando invisibili: fate però attenzione a non sprecare tutta l'energia...

Ogni stanza è numerata (quella della principessa è la numero 149) e parecchie hanno un nome. Nel caso si voglia iniziare un successivo turno di gioco dall'ultima stanza raggiunta, introdurre il suo nome all'inizio, quindi batti RETURN.

Idea	8
Giocabilità e suono	8
Animazione	8
Voto	8

1985 The day after

Far decollare una navicella può rappresentare una operazione molto difficile. Nel caso di 1985, ultimo uscito tra i giochi della Mastertronic, la cosa è forse ancora più complicata. Intendiamoci: la navicella è perfettamente manovrabile, un fascio di energia permette di salire verso l'alto e spostarsi, mentre il joystick da la giusta direzione. Ma il passaggio è talmente stretto che... ci vuole un po' di pratica.

Lo scenario è postatomico. E' il giorno dopo: il grande fratello del romanzo di Orwell (1984) è stato sconfitto. La terra deve imparare a sopravvivere da sola. I territori che vi dividono dalla salvezza (rappresentata da dosi di plasma nucleare) sono in tutto 12.

Gli ostacoli sono molti, i pericoli anche. La grafica è essenziale ma efficace. Ciò comunque (come ben sappiamo) consente una bella velocità di gioco.

Da provare subito.

Idea	9
Giocabilità	7
Animazione	8
Voto	8

THE PROFESSIONAL COMPUTERING
Olympia IL COMPUTER
HARD - SOFT - CONSULENZA
PROCEDURE: Multiutenza Multiposto
GESTIONALE: Integrato Vendite
Contabilità Paghe
Word Processing
SOLUZIONI
chiavi in mano
samufficio
Via Melzo, 8 - Milano - Tel. 2715779

SEIKOSHA

NON AVRAI ALTRA STAMPANTE

Seikosha ti invita nel meraviglioso mondo delle sue stampanti.
Un mondo fatto di progresso, di elevatissima qualità, velocità e silenziosità di stampa.
Seikosha oggi ti propone la piú vasta gamma di stampanti, nate per esaltare le prestazioni di ogni tipo di computer.
All'altezza di ogni esigenza, anche della tua che usi i Computer Commodore.
La tua necessità di stampa trova nel modello GP 500 VC, con 80 colonne e 50 caratteri al secondo, il miglior rapporto fra il prezzo, che è particolarmente contenuto, e le prestazioni di tutto rispetto.
Ma se hai delle applicazioni di Word Processing, solo GP 550 A con 80 colonne e 50 caratteri al secondo, anche Near Letter Quality a 25 caratteri al secondo, si impone per le sue prerogative di macchina bivalente: stampa comune e produzione di documenti.
Se le tue necessità ti impongono l'uso del colore, scopri GP 700 VC che fà del colore un vero spettacolo, infatti con 80 colonne e 50 caratteri al secondo, consente la stampa in alta risoluzione di 7 colori base e un numero praticamente illimitato di sfumature.
Seikosha e Commodore: una coppia che và d'amore e d'accordo.

SEIKOSHA
Distribuzione esclusiva: GBC Divisione Rebit

Il principale fascino degli *adventures* è quello di creare un mondo totalmente nuovo e sconosciuto, all'interno del quale sono disseminati tesori e pericoli... ma chi si avventura per la prima volta in un luogo, corre il rischio di perdersi, o di non saper trovare la strada del ritorno.

Inoltre, in questo tipo di giochi, può succedere di dimenticarsi di andare a visitare dei luoghi che potrebbero, invece, contenere oggetti indispensabili a completare l'*adventure*.

Se poi si aggiunge la necessità di ricordarsi le particolari frasi che servono a superare le più strane situazioni (per poter riprendere in seguito il gioco), ecco che risulta indispensabile tenere un promemoria di ciò che stiamo facendo.

In poche parole dobbiamo *mappare* l'adventure.

Le tecniche che si possono utilizzare per raggiungere i risultati desiderati sono molte, e possono anche rispondere a gusti personali di grafica e praticità.

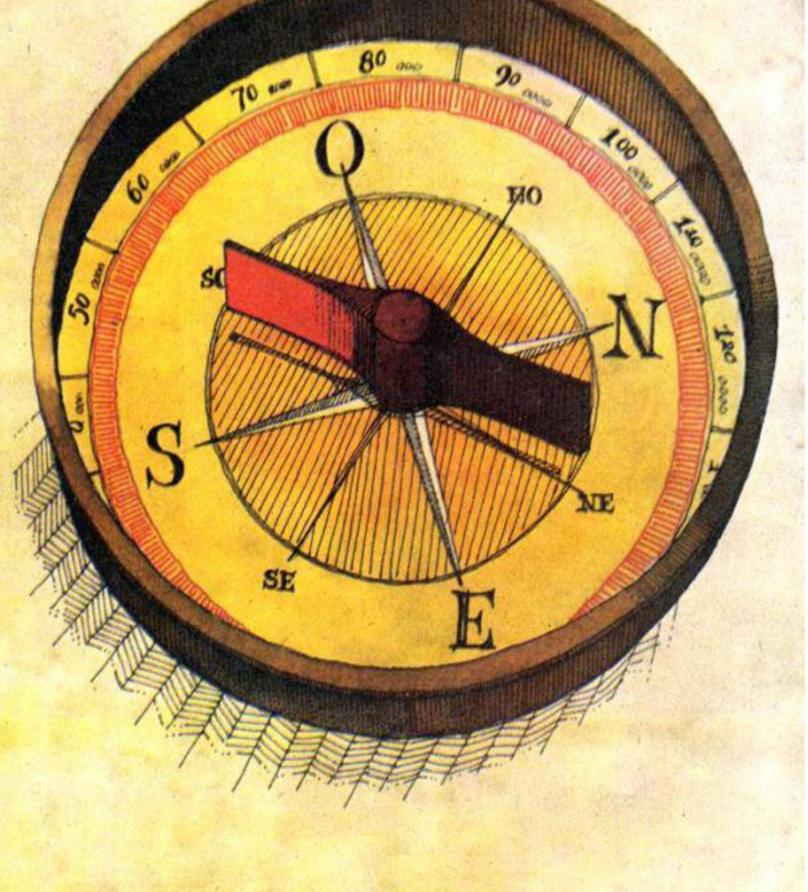
Per prima cosa, cerchiamo di capire quali sono gli elementi che dobbiamo tener presenti per ogni luogo dell'adventure:

descrizione del posto;

vie di uscita (ed eventuali frasi particolari per poterle usare: es. *go door, jump, enter,* etc.);

oggetti presenti;

frasi particolari da usare (es. *push button, kill guard with axe,* etc.);

oggetti che bisogna avere o indossare per poter effettuare una certa azione in quel luogo (es. avere la chiave per poter aprire la porta, indossare un paio di guanti per poter raccogliere un braciere ardente).

Il sistema più usato per mappare è quello di disegnare dei grandi quadrati, ognuno rappresentante un luogo, e scrivere, all'interno di esso, tutti gli elementi che vi si trovano e/o che ci servono (fig. 1).

Per collegare tra di loro i luoghi, faremo partire da ogni lato del quadrato delle linee, nelle varie direzioni di uscita possibili, rispettando i punti cardinali geografici.

Quando ci recheremo nelle zone contigue, dovremo sempre verificare se vi è la possibilità di tornare nel luogo da dove si era partiti.

Se questo avviene, la strada sarà aperta in entrambe le direzioni, e potremo disegnare sulla linea due frecce, che indicano i due versi opposti.

Se il passaggio è permesso in un solo senso, disegneremo solo la freccia che indica il verso in cui è consentito il passaggio.

Nella maggioranza dei casi, il passaggio è transitabile nei due sensi, e si può quindi semplificare il disegno inserendo solo le frecce che indicano unidirezionalità.

Problemi di mappa possono sorgere quando, con su e giù *(up. down)*, si cambia il livello su cui si svolge l'azione. Queste direzioni possono essere indicate con una linea curva.

La descrizione del luogo deve essere molto sintetica e, a seconda delle preferenze, può es-

Carta, Penna e Bussola

Per orientarsi lungo il percorso di un adventure spesso è utile affidarsi a un metodo molto tradizionale: prendere appunti

sere scritta in inglese o tradotta in italiano.

La trascrizione degli oggetti presenti può essere integrale o limitata a quelli che interessano l'azione: per questi ultimi, è utile associare una parola o una breve fase che ricordi l'uso che ne deve essere fatto, o l'altro oggetto a cui va abbinato (es. *OPEN DOOR - RED KEY).*

Un consiglio: disegnare le mappe con una matita, per poter facilmente correggere gli inevitabili errori, frequenti soprattutto in quegli adventures, mal strutturati, in cui si ha sovrapposizione di luoghi. In alcuni posti, inoltre, le informazioni da inserire nel quadrato possono essere molte, e potreste essere costretti a ingrandire il quadrato stesso.

Questo metodo, anche se permettete un buon colpo d'occhio della zona che stiamo visitando, può presentare degli inconvenienti nelle situazioni più complicate.

Per ovviarli si può adottare un altro sistema, decisamente più pratico: rappresenteremo i vari luoghi con piccoli quadrati, uniti tra loro come visto precedentemente, all'interno dei quali scriveremo un numero progressivo per ogni nuovo luogo visitato.

Su un altro foglio, potremo riportare i riferimenti necessari per ogni stanza (fig. 2).

Questo metodo permette di avere più spazio per le informazioni che ci interessano, che risulteranno chiare e complete.

I labirinti e...

Ma non tutto è così semplice: molto spesso, negli adventures sono presenti dei labirinti.

Per labirinti di intendono quei luoghi in cui posti diversi hanno la stessa identica descrizione: non riusciremo quindi a capire se, digitando *NORD,* ci siamo in effetti recati in quella direzione o se siamo rimasti nel luogo di partenza.

Inoltre, nei labirinti a volte non vengono rispettate le direzioni geografiche, e le mappe ri-

sulteranno inevitabilmente contorte e all'apparenza inestricabili.

Per quanto complessi, esiste però un metodo per mapparli correttamente: il *metodo di Pollicino*.

Se è vero che tutti i luoghi di un labirinto hanno la stessa descrizione, è anche vero che l'eventuale presenza in uno di essi di un oggetto lo renderebbe, inequivocabilmente diverso dagli altri.

Si tratta quindi di raccogliere il maggior numero possibile di oggetti e di entrare nel primo luogo del labirinto. Qui lasceremo, per esempio, la spada.

Digiteremo quindi NORD. Se in questo nuovo luogo, descritto in modo uguale, al precedente, troveremo una spada, vuol dire che in effetti non ci siamo mossi e che NORD è una falsa direzione.

Se non vi è la spada, allora ci troveremo in un luogo diverso, lasceremo un altro oggetto e continueremo così sino a completare l'intera mappa del labirinto.

È importante visitare tutti i luoghi. In *Message from Andromeda*, per esempio, in una stanza di un lungo e complicato labirinto, vi è un muro su cui è scritta una parola, indizio indispensabile per risolvere l'adventure.

Un esempio di labirinto mappato si può vedere in fig. 3.

Nei prossimi numeri metteremo in pratica quanto imparato, risolvendo totalmente un adventure, in modo da aiutare chi non si è mai avvicinato a questo genere di giochi o chi non è mai riuscito a completarne uno.

Arrow of Death (part 1)

1 - palazzo
(messaggero, amuleto)
examine messenger
get amulet
examine amulet

2 - palazzo

3 - stanza del trono
(Zardra il mago,
bacchetta d'oro)
examine Zardra
NON esaminare la Golden Baton

4 - cucina
(gancio)
examine kitchen
get door

5 - camera del re
(letto, armatura,
stemma, cucina)
get armour
examine bed
get pillow
turn coat (3 volte)
(per la 6 GO PASSAGE)

6 - stanza segreta
(spada)
get sword
cut pillow
(con la spada)
get purse
examine purse
drop purse
drop pillow
get coin

Wax Works

Per qualsiasi quesito, richiesta di aiuto, idea o collaborazione inerente gli adventure (e solo questi!), scrivete direttamente all'autore. Le lettere più interessanti riceveranno risposta

Giuliano Boschi - via F. Massi 12 - 00152 Roma.

E' PRIMAVERA, APRI UNA FINESTRA NELLO SCHERMO

In un programma, o durante la sua esecuzione, capita spesso di dover conservare alcune frasi sullo schermo, mentre è necessario cancellare tutto il resto parecchie volte. Ne risultano complicazioni che sarebbe meglio scrivere con un pennarello direttamente sullo schermo del TV. Invece di agire sul televisore, procurandovi magari qualche rimprovero dai familiari, vi diremo come ottenere lo scopo mediante un programma che funziona da "fissativo" per quelle indispensabili scritte.

Il programma pubblicato in queste pagine, insieme con alcune routine dimostrative, è quello che fa per voi. Rende infatti "indelebili" tutte le frasi, messaggi e scritte che volete in modo più versatile delle funzioni Windows disponibili su altri computer.

Per i tuoi giochi o per le applicazioni più serie, ecco come simulare la funzione window sul Commodore 64.

Come si usa

Dopo avere scritto e registrato (ottima precauzione) il programma, fatelo partire e, se non ci sarà segnalazione d'errore, sarete pronti per la seguente prova:

- cancellate lo schermo con SHIFT e CLR-HOME;
- premete 3 volte il tasto RETURN;
- premete la barra spaziatrice *contemporaneamente* col tasto SHIFT;
- scrivete la frase: "QUESTA SCRITTA E' INDELEBILE", badando di premere la barra dello spazio insieme con lo SHIFT;
- premete, alla fine della frase, ancora la barra dello spazio con lo SHIFT;
- premete SHIFT e RETURN due volte;
- scrivete
  SYS 49296: SYS 49152: POKE 720,1
...e premete RETURN.

Vedrete la frase digitata cambiare di colore e restare al suo posto anche se provate a cancellare lo schermo (Shift + Clr/home).

Scrivete ora
POKE 720,0

La frase diventerà nera: le scritte, infatti, assumono, il colore specificato da POKE 720,X in cui X è il codice del colore.

Fate un'altra prova: spostatevi col cursore in un'altra parte dello schermo e digitate un'altra frase che inizi e finisca con lo spazio premuto con lo SHIFT. Digitate SYS 49296: anche questa scritta diventerà una "window", ossia una "finestra" all'interno della quale nulla possono le normali operazioni di cancellazione.

Per creare finestre estese in verticale basta allineare verticalmente varie scritte

```
WINDOWS

  10 REM  ** GESTIONE WINDOWS **
  20 REM  ** SUL COMMODORE 64 **
  30 REM  ** BY FABIO SORGATO **
  50 :
  60 REM  -- SYS 49152 ABILITA
  70 REM  -- SYS 49258 DISABILITA
  80 REM  -- SYS 49296 CREA NUOVE
     WINDOWS
  90 REM  -- POKE 720,X COLORE WIN
     DOWS
  91 :
 100 FOR K=0 TO 178:READ A:CK=CK+A
     :POKE 49152+K,A:NEXT
 110 IF CK<>23221 THEN PRINT"ERROR
     E NEI DATI.":END
 120 PRINT"WINDOWS ATTIVATE !"
 130 :
1000 DATA  120,169,17,141,20,3
1010 DATA  169,192,141,21,3,88
1020 DATA  169,0,133,2,96,165
1030 DATA  2,41,7,170,234,189
1040 DATA  128,192,133,251,133,253
1050 DATA  189,138,192,133,252,189
1060 DATA  133,192,133,254,160,0
1070 DATA  177,251,170,201,96,208
1080 DATA  9,165,2,73,128,133
1090 DATA  2,76,64,192,165,2
1100 DATA  41,128,240,18,138,145
1110 DATA  253,166,254,138,24,105
1120 DATA  212,133,254,173,208,2
1130 DATA  145,253,134,254,200,192
1140 DATA  200,208,211,230,2,165
1150 DATA  2,41,7,201,5,208
1160 DATA  6,165,2,41,0,133
1170 DATA  2,76,49,234,120,169
1180 DATA  49,141,20,3,169,234
1190 DATA  141,21,3,88,96,0
1200 DATA  4,5,6,7,196,196
1210 DATA  197,198,0,200,144,88
1220 DATA  32,4,4,5,6,7
1230 DATA  196,196,197,198,199,0
1240 DATA  120,160,0,152,170,133,2
     51
1250 DATA  133,253,169,4,133,252
1260 DATA  169,196,133,254,177,251
1270 DATA  145,253,200,208,249,230
1280 DATA  252,230,254,232,224,4
1290 DATA  208,240,88,96
```

```
UTILITY WIND

   10 REM  ** GESTIONE WINDOWS **
   20 REM  ** SU  COMMODORE 64 **
   30 REM  **  UTILITY  BASIC  **
   31 :
63000 Y9$="": FOR  I =1  TO  25: Y9
      $=Y9$+CHR$(17): NEXT
63005 FI=160:ER=0
63010 FOR  E9=1  TO  80:F9$=F9$+CHR
      $(32)+CHR$(32)+CHR$(32):NEXT
63019 REM  TAVOLA MEMORIA SCHERMO
63070 X=X-1:Y=Y-1:LX=LX-1:LY=LY-1
63071 IF  X+LX>39  OR  X<1  OR  LX<
      0  THEN  ER=1:RETURN:REM  CON
      TROLLO X
63080 IF  Y+LY>24  OR  Y<0  OR  LY<
      0  THEN  ER=1:RETURN:REM  CON
      TROLLO Y
63085 IF  EL>9  OR  EL<1  THEN  ER=
      1:RETURN:REM  CONTROLLO ELEME
      NTI DI FR$(X)
63086 SYS49258
63090 FOR  E9=1  TO  EL :CK=CK+LEN(
      FR$(E9)):NEXT
63100 FOR  E9=EL+1  TO  9:FR$(E9)=F
      9$:NEXT:EL=9
63110 A9=X:B9=Y:PRINT"[HOME]"LEFT$(
      Y9$,B9);SPC(A9-1);CHR$(FI)
63120 FOR  E9=1  TO  EL
63130 FOR  K9=1  TO  LEN (FR$(E9))
63140 PRINT"[HOME]";LEFT$(Y9$,B9);S
      PC(A9);MID$(FR$(E9),K9,1);CHR
      $(145)
63150 A9=A9+1
63151 IF  A9>X+LX  THEN  GOSUB  632
      00
63160 IF  B9>Y+LY  THEN  63180
63170 NEXT:NEXT
63180 SYS49296:SYS49152:RETURN
63200 PRINT"[HOME]"LEFT$(Y9$,B9);SP
      C(A9);CHR$(FI)
63210 B9=B9+1:A9=X: IF B9<=Y+LY THEN
       PRINT"[HOME]"LEFT$(Y9$,B9);S
      PC(A9-1)CHR$(FI)
63220 RETURN
```

sempre precedute e seguite dallo spazio con lo SHIFT.

Per cancellare le windows sarà sufficiente disabilitarle con SYS49258 e cancellare lo schermo, dopodichè crearne di nuove con SYS49296.

E' bene sottolineare che le windows, ed il loro contenuto, rimangono memorizzate anche se le disabilitate. Ciò vuol dire che è possibile disabilitarle con SYS49258 e, per riottenerle, limitarsi soltanto a digitare SYS49152.

Per creare windows all'interno di un programma basterà fare precedere e seguire la window da CHR$(160) usando, per esempio, il seguente formato:

```
10 PRINTCHR$(160);
"MESSAGGIO";CHR$(160)
```

Se invece siete più esigenti usate l'utility-scritta interamente in basic pubblicata in queste pagine: serve per creare windows con l'estensione orizzontale e verticale e nel punto dello schermo desiderato. Scrive, inoltre, all'interno della window il messaggio desiderato andando automaticamente a capo come se si trattasse di un piccoloschermo indipendente.

Usare la subroutine è molto semplice: basta definire alcune variabili e richiamare in seguito la subroutine. Se la finestra è troppo piccola in proporzione alla frase, il messaggio verrà troncato alla fine della window. Le variabili da definire sono:

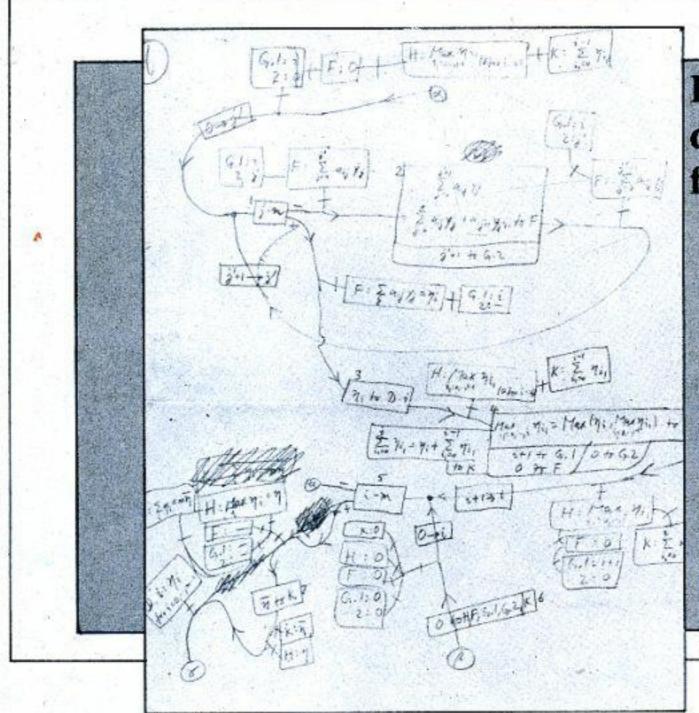
X= posizione orizzontale di inizio della window (da 2 a 38);
Y= posizione verticale di inizio della window (da 1 a 25);
LX= estensione orizontale della window (da 1 a 39-X);
LY= estensione verticale della window (da 1 a 24-Y);
EL= numero degli elementi usati del vettore FR$(K) (da 1 a 9);
FR$(k)= vettore in cui è contenuta la frase (max 255 caratteri).

Si è fatto ricorso ad un vettore, dato che una singola variabile stringa contiene al massimo 255 caratteri, che potrebbero risultare insufficienti nei casi di window di grandi estensioni. Se la vari abile ER diventa uguale a 1, vuol dire che non si sono rispettate le dimensioni consentite per la window. La subroutine viene richiamata con GOSUB63000.

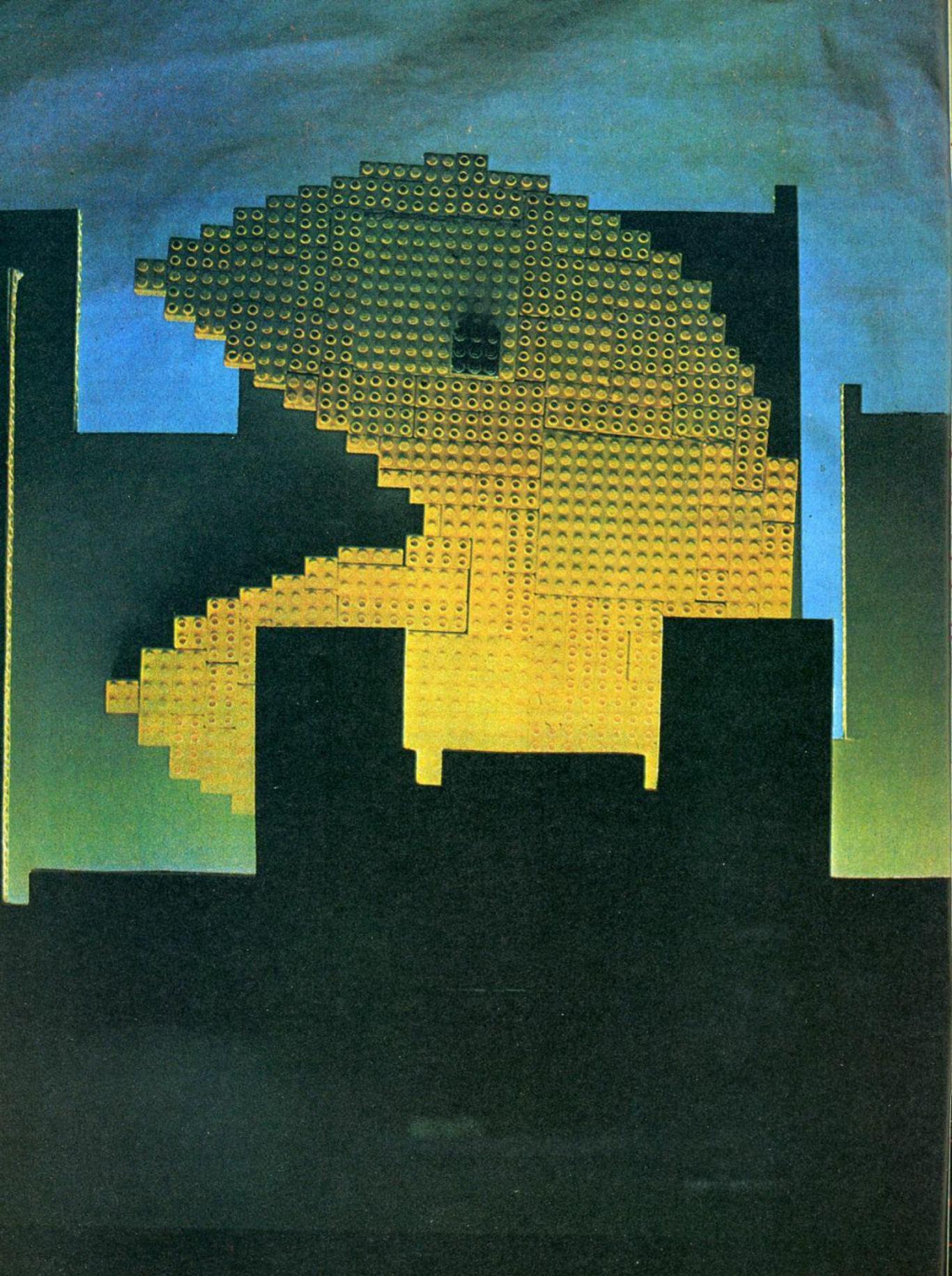
E' anche pubblicato un programmino d'esempio per l'uso dell'utility descritta.

Dato il RUN, il programma chiederà il numero delle frasi da scrivere. Si introdurrano, pertanto, le frasi desiderate. In seguito si dovrà introdurre la coordinata X di partenza della window, l'estensione orizzontale, la coordinata Y (che consiglio superiore a 10, altrimenti si sovrapporrà agli INPUT per la creazione di un'altra finestra), l'estensione verticale e il colore.

Creata in tal modo la window (a patto che non sia segnalato un errore) basterà premere un tasto per aggiungerne un'altra. Per uscire dal programma si ricorre al solito RUN STOP/RESTORE mentre, per fare riapparire le finestre create, occorre dare SYS49152.

Fabio Sorgato

Le immagini di questo fascicolo

1946. Un appunto di von Neumann sulla programmazione degli elaboratori elettronici. Al genio matematico di von Neumann si deve anche lo sviluppo della cosiddetta "teoria dei giochi", che si propone di definire in termini matematici la migliore strategia da seguire in qualunque tipo di competizione, dal gioco del poker ai problemi economici e produttivi. La teoria dei giochi è alla base di quell'insieme di tecniche che vanno sotto il nome di "ricerca operativa" e che si propongono di risolvere razionalmente e scientificamente ogni problema decisionale.

FACCIAMO UN VIDEOGIOCO

Avete in mente la sceneggiatura di un video-game ma non l'esperienza necessaria per "trasportarlo" sul computer? Dall'idea base fino al gioco completo, Commodore Computer Club vi porta per mano dietro le quinte del vostro prossimo videogioco.

Ma come si fa, partendo da un'idea, a costruire un gioco?

La cosa è molto più semplice di quello che possa sembrare in un primo momento, e questo articolo vuole appunto spiegare come si fa. Partendo da un'idea base, sviluppare un gioco, da soli! Tutto quello che segue fa riferimento al C64. Per il C16/PLUS4, prima di iniziare a leggere, guardate le note del riquadro.

Il gioco base

L'idea scelta per il nostro gioco è semplice: siamo in un labirinto pieno di ostacoli e vogliamo raggiungere il nostro compagno, che sta dalla parte opposta dello schermo. Se si va contro un ostacolo il gioco finisce. La prima idea che può venire in mente è quella di "mettere assieme" qualche riga basic per generare il labirinto, magari casuale, tanto per fare le cose più semplici. Provate a digitare le linee seguenti esattamente come sono. A queste, d'ora in poi, vanno aggiunti i vari pezzi (segmenti) di programma in modo da ottenere alla fine il programma completo.

```
160 POKE 53280,2:
    POKE53281,2:
    PRINT CHR$(158);
    CHR$(147)
180 SC=1024
190 CL=55296
200 N=50
220 FOR K=1 TO N
230 R=INT(RND(1)*1000)+1
240 POKE SC+R,160:POKE CL+R,1
250 NEXT K
```

Prima di spiegare il funzionamento di queste linee vediamo come funziona la memoria di schermo.

Il 64 ha 40 colonne e 25 righe per un totale di 1000 caratteri in una schermata; serviranno quindi 1000 byte di memoria per contenere i caratteri. Questi 1000 byte nel 64 sono normalmente posti a partire dalla locazione 1024 fino alla 2023ma.

Quando vogliamo stampare qualcosa in queste locazioni dobbiamo dare un'istruzione del tipo 'POKE loc,codcar', dove "loc" è la locazione di schermo (compresa tra 1024 e 2023) e "codcar" è il codice del carattere da stampare (i codici di tutti i caratteri sono riportati sul manuale del 64).

Per alcuni esemplari di Commodore 64, però, stampando con le POKE viste prima il carattere, questo non si vede perchè viene stampato dello stesso colore dello sfondo. Bisogna quindi colorarlo per renderlo visibile. Esiste nel 64 un'altra area di memoria, lunga anch'essa 1000 byte (da 55296 a 56295), che serve appunto per memorizzare il colore di

ogni carattere stampato.

La sintassi è la stessa che abbiamo visto per la memoria di schermo, cioè 'POKE loc,col' dove "col" è il colore , che può variare da 0 a 15. Sul manuale del 64 è riportata anche la tabella dei colori. E' su questi principi che si basano le routine di stampa del gioco.

Ed ora vediamo di analizzare in dettaglio le righe digitate prima: la prima **(160)** cambia i colori del bordo, dello sfondo, dei caratteri (CHR$158) e pulisce lo schermo (CHR$147). Poi, alle linee **180-190** vengono inizializzate le variabili SC e CL rispettivamente all'inizio della memoria di schermo e di colore.

Quindi **(linea 200)** viene posto il numero di ostacoli da stampare (=50) e, **(linea 220)** viene iniziato un ciclo che va da 1 al numero di ostacoli (cioè N).

La **linea 230** genera un numero casuale tra 1 e 1000, che verrà poi sommato alle variabili SC e CL per stampare in una posizione casuale dello schermo uno spazio in campo inverso colorato di bianco **(linea 240).** La **linea 250,** infine, chiude il ciclo.

Abbiamo così realizzato una breve routine in basic (provate a dare RUN) che genera sullo schermo un labirinto con 50 ostacoli casuali. E va bene, ma... se avessimo bisogno di un numero di ostacoli diverso da 50? Perchè non rendere la routine flessibile, in modo da poter introdurre da tastiera il numero di ostacoli da stampare? Potremmo sostituire la linea 200 con questa:

```
210 INPUT "NUMERO
OSTACOLI"; N
```

Questa linea, al contrario della precedente, che imponeva N=50, chiede (INPUT) quanti ostacoli vogliamo stampare. Alla domanda dovremo introdurre il numero seguito dalla pressione del tasto RETURN.

Però ora lo schermo si è "sporcato" di nuovo con il messaggio dell'INPUT e dovremo cancellarlo aggiungendo una linea:

```
210 PRINT CHR$(147)
```

Questa, come abbiamo visto prima, cancella lo schermo. Provate a dare RUN varie volte alla routine così modificata e cambiate il numero di ostacoli ogni volta, così da vederne i diversi effetti.

Dopo aver fatto questo, la prossima cosa che può venire in mente è quella di stampare il giocatore ed il bersaglio da raggiungere.

```
290 POKE SC,81:POKE CL,0
300 POKE SC+999,87:POKE
CL=999,5
```

La **linea 290** stampa nella prima locazione di schermo una pallina piena (codcar=81) che sarà il bersaglio, colorata di nero (col=0), mentre la **linea 300** stampa il giocatore, che è una pallina vuota (codcar=87) nell'ultima locazione di schermo (prima+999) colorata di verde (col=5).

Dando il run al programma ottenuto, si nota però che le due palline stanno ferme e che la scritta READY compare in alto allo schermo, bisogna perciò "inventare" qualcosa che ci permetta di muovere il nostro giocatore in modo da poter raggiungere il bersaglio.

Si potrebbe utilizzare un'istruzione del tipo:

GET A$:IF A$="car" THEN...

che rileva il tasto premuto. Ma ho preferito utilizzare la locazione 197 che rileva il codice del tasto premuto, un po' più complicata da gestire ma molto più efficiente.

Aggiungete queste linee al programma:

```
340 TI$="000000"
350 R=24:C=39
370 P=PEEK(197):IF P=64THEN 370
400 IF P=10 THEN C=C-1:REM
    COD. TASTO "A" (SINISTRA)
402 IF P=18 THEN C=C+1:REW
    COD. TASTO "D" (DESTRA)
404 IF P=9 THEN R=R-1:REM COD.
    TASTO "W" (ALTO)
406 IF P=23 THEN R=R+1:REM
    COD. TASTO "X" (BASSO)
480 POKE SC+R*40+C,87:POKE
    CL+R*40+C,5
490 GOTO 370
```

La **linea 340** azzera il tempo utilizzando la variabile stringa TI$ nel modo descritto sul manuale del computer. Non avevamo stabilito di realizzare un gioco a tempo? La **linea 350** inizializza le variabili R(riga) e C(colonna) alla posizione iniziale del giocatore, cioè riga 24 e colonna 39. La **linea 370** controlla con un PEEK(197) se è stato premuto un tasto. In caso negativo (codice64= nessun tasto premuto) ritorna su sè stessa e aspetta la pressione di un tasto senza proseguire nel programma.

In caso che si prema un tasto, le **linee da 400 a 406** ne controllano il codice che, nel caso sia uno di quelli specificati, modifica la colonna e la riga della posizione del giocatore. I codici 10, 18, 9, 23 corrispondono rispettivamente ai tasti "A", "D", "W", "X" per le direzioni sinistra, destra, alto e basso.

La **linea 480** stampa il giocatore nella nuova posizione secondo la formula:

Posizione=SC+riga*40t colonna

e lo colora di verde secondo lo stesso principio. La **linea 490** infine fa ritornare il programma sulla linea 370, la quale controlla di nuovo se...ecc. ecc.

Correzioni

Facendo girare il programma digitato, si nota però che ha dei difetti. Il primo, subito rilevabile, è che quando il giocatore si muove, lascia dietro di sè una "scia", questo perchè noi stampiamo il giocatore nella nuova posizione senza prima cancellarlo dalla vecchia, quindi dovremo aggiungere una linea del tipo:

```
390 POKE SC+R*40+C,32
```

che cancella (carcod 32=spazio vuoto) il giocatore prima di spostarlo, ottenendo così il nostro scopo.

Il secondo difetto, meno evidente ma

molto più pericoloso, è che il programma non controlla i limiti di spostamento!!!

Se voi provate a muovervi continuamente a destra o a sinistra fino ad uscire dallo schermo da un lato, noterete che il giocatore non si ferma sul bordo dello schermo come sarebbe logico, ma continua ad avanzare fino a che non rilasciate il tasto.

Se questo per molti potrebbe non essere un difetto, lo diventa quando invece di "andare fuori" dallo schermo da destra o da sinistra lo si fa dall'alto o dal basso (non fatelo!).

Nel modo descritto, infatti, si invadono zone di memoria "pericolose" (il programma basic da una parte e i puntatori di sistema dall'altra) rischiando di "perdere" il programma in memoria oppure, peggio, di ottenere il blocco (crash) del sistema con la conseguenza di dover spegnere la macchina e ricominciare da capo a digitare.

In effetti il giocatore potrebbe stare attento a non oltrepassare i bordi dello schermo, ma per sicurezza è meglio inserire nel programma opportune linee di controllo che, anche se rallentano leggermente l'esecuzione, evitano spiacevoli conseguenze. Le linee in questione sono:

```
420 IF C<0 THEN C=0
430 IF R<0 THEN R=0
440 IF R>24 THEN R=24
450 IF C>39 THEN C=39
```

La **linea 420** controlla se la colonna di spostamento è inferiore a 0 (infatti se siete in colonna 0 e spostate a sinistra, la colonna diventa -1, andando così fuori dallo schermo) e, in tal caso, la pone uguale a 0, non permettendo così di spostarci oltre il limite inferiore.

Analogo discorso va fatto per la riga **(linea 430).** Le **linee 440 e 450** fanno la stessa cosa delle due precedenti, solo che invece di controllare i limiti minimi controllano quelli massimi. E' importante sottolineare che questi controlli vanno fatti prima di spostare il giocatore perchè sarebbe illogico spostarlo fuori dallo schermo (con tutte le conseguenze viste prima) e poi controllare se lo si è spostato fuori.

Analogo discorso va fatto per la riga **(linea 430).** Le **linee 440 e 450** fanno la stessa cosa delle due precedenti, solo che invece di controllare i limiti minimi controllano quelli massimi. E' importante sottolineare che questi controlli vanno fatti prima di spostare il giocatore perchè sarebbe illogico spostarlo fuori dallo schermo (con tutte le conseguenze viste prima) e poi controllare se lo si è spostato fuori.

Facendo girare il programma così modificato si ottiene una discreta versione del gioco che però (non finiscono mai) ha un altro difetto.

Infatti si noterà che nel muoversi, se si urta un ostacolo o un bersaglio... non succede niente, ma semplicemente "gli passiamo sopra" cancellandolo. A questo punto si potrebbe pensare: "Tutta questa fatica per un gioco senza difficoltà?" Niente paura, basta infatti aggiungere una linea che controlli se il giocatore urta un carattere-ostacolo, cioè una linea come questa per esempio:

```
470 IF PEEK(SC+R*40+C) < >
32 THEN PRINT CHR$(147);TI$:END
```

La **linea 470** controlla con un PEEK mediante la formula vista prima la posizione che dovrà assumere il giocatore prima di essere stampato, e se questa è uno spazio vuoto (carcod=32) continua nell'elaborazione, altrimenti stampa il tempo di gioco e termina l'esecuzione.

Questa linea potrebbe anche andare bene. Se non fosse che il programma si ferma allo stesso modo sia che toccate un ostacolo sia che toccate il bersaglio, dato che non abbiamo scritto nessun commento, ma solo il tempo di gioco (controllare per credere).

Ora, la persona che sta giocando sa (o almeno dovrebbe sapere se non era troppo distratta) se è andata contro ad un ostacolo o se ha raggiunto il bersaglio, ma visto che abbiamo a disposizione un calcolatore (e pure fantastico come il C64), perchè non facciamo fare tutto a lui?

Cambiamo la linea 470 in:

```
470 IF PEEK(SC+R*40+C) < >
32 THEN P=PEEK(SC+R*40+C):
GOTO 620
```

e aggiungiamo le seguenti linee:

```
620 TM$=TI!:PRINT CHR$(147)
640 IF P=81 THEN PRINT "HAI
    RAGGIUNTO IL BERSAGLIO"
650 IF P=160 THEN PRINT
    "HAI URTATO
    UN OSTACOLO"
660 PRINT" TEMPO:"TM$
670 END
```

La **linea 470** così modificata controlla sempre cosa c'è nella posizione che dovrà assumere il giocatore. E nel caso non sia uno spazio, memorizza il contenuto della variabile P e passa alla **riga 620.** Questa linea memorizza il tempo di gioco nella variabile TM$, questo perchè TI$ viene incrememtata automaticamente da computer e se prima di stampare il tempo dobbiamo compiere operazioni particolarmente "lunghe" il risultato del gioco viene falsato. Se invece memorizziamo subito il tempo in una variabile a nostra scelta (nell'esempio riportato: TM$) che il computer non "tocca", non abbiamo più problemi.

La **linea 640** controlla se P (P era il codice della "cosa" contro cui abbiamo urtato) è uguale ad 81, che, guarda caso, è proprio il codice del bersaglio; in questo caso stampa un opportuno messaggio.

In **linea 650** invece si controlla se questo codice era quello di un ostacolo (quadratino bianco, cioè spazio in campo inverso codcar=160) e, in caso affermativo, stampa un messaggio diverso (HAI URTATO UN OSTACOLO).

La **linea 660** stampa il tempo di gioco e infine la **linea 670** termina l'esecuzione del programma.

Più difficile ancora

Il programma, così com'è, potrebbe andare bene ma solo per un numero abbastanza basso di ostacoli; se invece vogliamo complicare la partita inserendo un numero alto (maggiore, per esempio, di 250) cominciano a sorgere dei problemi. Può infatti capitare che il bersaglio sia circondato da ostacoli oppure che siamo in un vicolo cieco che, anche con tutta la buona volontà, non si riesce a superare.

Si potrebbero quindi inserire nel programma delle linee che premendo un tasto, cancellano tutti gli ostacoli intorno liberandoci la strada.

Le seguenti linee possono servire allo scopo:

```
380 IF P=1 THEN GOSUB 520
520 S=SC+R*40+C
540 IF S<1984 THEN POKE S+40,32
550 IF S>1064 THEN POKE S-40,32
560 IF S<2023 THEN POKE S+1,32
570 IF S>1025 THEN POKE S-1,32
580 FOR K=1 TO 1000:NEXT
590
RETURN
```

La **linea 380** (inserita quindi nel blocco di programma che controlla la pressione dei tasti) controlla se viene premuto il tasto RETURN (PEEK 197=1) e, in caso affermativo, passa il controllo al sottoprogramma che inizia dalla **linea 520.**

Questa linea memorizza la posizione attuale del giocatore nella variabile S, le **linee seguenti** eseguono dei controlli di limite (a cosa servono l'abbiamo visto in precedenza) e quindi se tutto va bene cancellano gli ostacoli. Per cancellare gli ostacoli tutti intorno si dovrà cancellare la posizione attuale del giocatore (S)+1 (POKE S+1,32), quella precedente (POKE S-1,32), quella sopra (ricoradate che lo schermo ha 40 colonne) che sarà P-40 (POKE P-40,32) e quella sotto (POKE P+40,32).

Ovviamente nè P-1, nè P+1, P-40 oppure P+40 dovranno dare un risultato minore di 1024 o maggiore di 2023 ed ecco il perchè dei controlli di limite (IF S> eccetera).

Con questo sistema vengono cancellati solo gli ostacoli che stanno "dentro" lo schermo, ignorando le locazioni adiacenti al giocatore ma "fuori" dello schermo visibile.

La **linea 580** crea un ciclo di ritardo di circa un secondo, perchè non sarebbe giusto usufruire di questa utile funzione senza avere qualche handicap (in questo caso la perdita di tempo).

La **linea 590** infine ritorna alla linea subito dopo quella che ha chiamato il sottoprogramma (la linea dopo il GOSUB 520).

Per i più pigri... abbiamo finito !!! (finalmente). Il gioco infatti funziona così com'è e non ha più difetti. Per i più intra-

prendenti si può anche sofisticare maggiormente il gioco aggiungendogli musichette, effetti speciali.

Ed è infatti quello che faremo nella seconda parte di questo (lunghissimo) articolo. In figura è pubblicato il diagramma a blocchi semplificato del programma, che illustra le principali operazioni che il programma esegue.

Piccoli giochi crescono: i suoni

I mini miglioramenti che vengono in mente riguardano sicuramente gli effeti sonori.

Costruiamo una routine basic che generi il rumore di una esplosione quando si va contro ad un ostacolo.

Per prima cosa, dobbiamo cancellare dal programma precedente le **linee 620, 640, 650, 660, 670,** semplicemente scrivendo il solo numero di linee e premendo RETURN. Dopo aver cancellato queste linee, dobbiamo aggiungere le seguenti: **620, 640, 660, 680, 690, 710, 730, 740, 750, 760, 770, 800, 810, 815, 820.**

La **linea 620,** vedere nel listato, memorizza il tempo; la **linea 640** controlla se l'oggetto che abbiamo colpito è il bersaglio (carcod=81) e in caso affermativo passa il controllo alla linea 800, dove viene stampato un messaggio di vittoria e terminato il gioco.

In caso che l'oggetto colpito non sia il bersaglio, il programma continua con la **linea 660,** dove vengono azzerati tutti i registri musicali, che vanno appunto dalla locazione 54272 alla 54296. E' opportuno inserire una linea del genere in ogni programma che debba suonare qualcosa, in modo da non lasciare i registri "sporchi" ed eventualmente causare anomalie nel funzionamento del programma.

Nelle **linee 680/690** vengono settati alcuni valori: la locazione 54298 controlla il volume del suono che può variare da 0 (minimo) a 15 (massimo), le due locazioni 54272/54273 settano il valore della nota da suonare in questo modo: 54272 contiene il valore basso della nota mentre 54273 il valore alto.

Ma cosa sono questi valori: "alto" e "basso"? Un suono può essere "indicato" da una frequenza, che indica appunto l'altezza del suono. L'unità di misura per la frequenza è l'Hertz (Hz) e può spaziare (nelle frequenze udibili) da circa 16 a 16000 Hz. Il 64 vuole le note appunto in Hz, ma come sappiamo in una locazione di memoria non si può inserire un valore superiore a 255, quindi si dovrà trovare un'altro modo per ovviare all'inconveniente.

Il modo più semplice è quello di dividere la frequenza della nota per 256, l'intero risultato ottenuto sarà appunto il cosiddetto valore byte "alto" da POKare nella locazione 54273, mentre il resto della divisione sarà il valore basso da inserire nella locazione 54272.

Per fare un esempio: ammettiamo che la nota che dobbiamo suonare abbia la frequenza di 4097 Hz e dividiamo appunto 4097 per 256. Il risultato della divisione è 16 ed il resto è 1. 16 è il valore alto mentre 1 diventa il valore basso della nota. Per far suonare questa nota dovremo quindi inserire nel programma uan riga del tipo:

POKE54272,1:POKE54273,16.

Chiudendo questa parentesi e rimandando al manuale del 64 per ulteriori dettagli continuiamo la descrizione della routine.

La locazione 54277 controlla il ciclo di attack/decay della nota, mentre la 54278 ne controlla il sunstain/release. Nonostante Commodore Computer Club abbia già trattato in precedenza tali argomenti, diamo comunque una semplicissima descrizione di che cosa fanno.

Le locazioni musicali

Questi parametri controllano in quanto tempo il volume di una nota passa dal minimo al massimo e in quanto tempo il volume passa a fine nota dal massimo al minimo. Per fare un esempio, il suono di un volume cessa non appena si smette di "sfregare" la corda, mentre quello di un pianoforte anche se si è rilasciato il tasto "sfumando" a poco a poco. Sono appunto questi gli effetti che vengono controllati dai quattro parametri visti prima.

La locazione 54276 controlla la forma d'onda della nota, cioè il tipo di suono, che può variare da dolce (flauto) a squillante (clavicembalo). In questo caso però, il valore 129 POKato appunto in 54276 significa che abbiamo scelto come tipo di suono il rumore bianco, cioè un miscuglio di suoni causali che producono un effetto simile a quello di un'esplosione.

Le linee "musicali" sono finite, infatti il programma continua con la **710,** che contiene un ciclo FOR-NEXT per far variare ripetutamente il colore dello schermo, costruendo così un discreto effetto ottico.

Le linee da **730 a 750** stampano gli opportuni messaggi di fine gioco e la **linea 760** azzera, come visto prima, tutti i registri musicali.

Ma...cos'è quel misterioso "POKE198,0" nella **linea 770?**
Semplice: quando voi premete dei tasti durante lo svolgimento di un qualsiasi programma (in questo caso possono essere i tasti di movimento) questi vengono memorizzati in un "buffer di tastiera" cioè un pezzo di memoria dedicato alla tastiera. Il buffer viene "scaricato" (sul video) alla fine del programma, quando cioè viene restituito il controllo all'interprete basic e quindi tutti i caratteri presenti in esso vengono stampati, sporcando la schermata finale. La locazione 198 contiene il numero di caratteri contenuti nel buffer e quindi se noi la azzeriamo prima di terminare il programma, il buffer non verrà scaricato lasciando lo schermo inalterato.

Musica, musica!

Al nostro programma abbiamo, finora, aggiunto un semplice effetto sonoro, che contribuisce a vivacizzare il gioco. Perchè non proviamo ora ad aggiungerne un altro, questa volta in caso di vittoria, cioè di raggiungimento del bersaglio? Magari il secondo effetto invece di essere una semplice nota potrebbe essere

```
100 GOSUB 1070
800 RESTORE
810 FOR K=54272 TO 54296:POKE K,0:NEXT
840 POKE 54296,10:POKE 54276,65:POKE 54277,190
845 POKE 54275,15:POKE 54274,15
860 READ H,L,D:IF H=-1 THEN FOR K=54272 TO 54296:POKE
K,0:NEXT:GOTO 930
890 POKE 54273,H:POKE 54272,L:FOR X=D-50 TO D-20:POKE
54278,136:NEXT
900 FOR T=1 TO D:NEXT
910 POKE 54273,0:POKE 54272,0:POKE 54276,0:GOTO 840
930 REM ** FINE **
950 PRINT CHR$(147)
960 PRINT" BRAVISSIMO! HAI RAGGIUNTO IL BERSAGLIO!"
970 PRINT" TEMPO:";TM$
980 PRINT" ...CIAO":POKE 198,0:END
1070 S=0:FOR K=1 TO 39:READ A:S=S+A:NEXT
1090 READ A:IF S<>A THEN PRINT" ERRORE NEI DATI
MUSICALI!":END
1100 RETURN
1160
DATA34,75,250,43,52,250,51,97,375,43,52,125,51,97,250,57,172,
250,51,97
1170
DATA500,0,0,125,43,52,250,51,97,250,57,172,1000,51,97,1000,-1
,-1,-1
1180 DATA 6214
```

una (altrettanto semplice) musichetta.

Semplice, perchè una musica non è altro che una successione di singole note, e quindi basterà trovare la musica suonarle tutte, una alla volta, per ottenere la musica completa. Per fare questi si potrebbe utilizzare il programma di prima lievemente modificato. Si potrebbe cioè aggiungere una linea del tipo:
POKE54272,...:POKE54273,...
per ogni nota da suonare, ma nel caso di motivi lunghi il programma rischierebbe di diventare lungo come un trattato di medicina e voi di diventare vecchi prima di averlo finito.

La soluzione più semplice è quella di mettere le note in righe DATA, leggendone e suonandone una alla volta. Le linee seguenti aggiungono al programma una semplice musichetta in caso di vittoria.

Prima di digitarle però cancellate le **linee 800, 810, 815, 820.**

La **riga 100**, che abbiamo aggiunto all'inizio del programma (e quindi sarà la prima che esegue), esegue un GOSUB1070 in modo da andare a leggere i dati musicali per controllare se sono stati digitati correttamente. Le linee **da 1070 a 1100** servono appunto allo scopo: esse leggono i 39 dati e mettono in S la somma di tutti questi dati (**linea 1070**).

La **1090** legge il 40mo dato, che è appunto la somma dei dati se sono stati inseriti correttamente. A questo punto la stessa linea confronta appunto la somma esatta con il risultato ottenuto prima (S) se sono uguali significa che abbiamo inserito correttamente i dati, mentre se il confronto non ha successo abbiamo sbagliato qualcosa e il programma si ferma, permettendoci così di confrontare le righe che abbiamo scritto con quelle sulla rivista.

Se il confronto ha successo, la **linea 1100** fa riprendere il controlloal programm principale.

Nel caso che raggiungiamo il bersaglio il comando del programma passerà tramite la **linea 640** alla **linea 800**, che esegue RESTORE (che non ha niente a che vedere con l'omonimo tasto) che azzera il puntatore ai dati, cioè, quando noi leggiamo un dato con una READ il puntatore dei dati il puntatore dei dati si incrementa, permettendoci con un'altra READ di leggere il prossimo dato e così via.

Se noi leggiamo tutti i dati il puntatore si incrementa fino a puntare oltre la fine dei dati, e se noi eseguiamo un'altra READ tentiamo di leggere un dato che in realtà non esiste generando un messaggio di errore (OUT OF DATA). Il comando (e ripeto: il comando e non il tasto) RESTORE azzera questo puntatore in modo che ogni volta che viene chiamata la routine inizia a leggere dal primo dato inserito.

Le **linee da 810 a 845**, come visto in precedenza, azzerano i registri musicali e settano alcuni valori iniziali per la musica.

La **linea 800** legge i tre dati per la musica. Ma perchè tre, visto che come abbiamo detto, per suonare una nota sono necessari solo due valori? Perchè servono due valori per la frequenza, ma è

anche necessario un terzo valore che indica la durata della nota. La stessa linea controlla poi che il dato non sia un -1.

Il valore -1 infatti, che non potrà mai essere utilizzato per generare una nota (le frequenze non possono essere negative) dice al programma che la musica è finita. In tal caso il programma continua azzerando i registri musicali e saltando alla **linea 930**, dove vengono stampati i soliti messaggi di fine programma, azzerato il buffer di tastiera e terminata l'esecuzione.

Nel caso invece che il valore letto non sia -1, il programma continua leggendo e POKando nei registri 54272/54273 i valori della nota. La **linea 900** genera un ciclo di ritardo pari appunto alla durata della nota (il terzo dato letto).

Una particolarità del SID (il generatore sonoro del 64) è che prima di suonare una nuova nota bisogna azzerare sia i due registri del valore alto/basso che il registro della forma d'onda.

E' per questo che è necessaria **la linea 910**, oltre che per far ritornare il programma alla **linea 840**, dove rimette la forma d'onda giusta, legge, suona eccetera, e così via fino alla fine della musica.

Siamo così riusciti ad aggiungere una musichetta al programma, che ora è diventato piuttosto complesso. Perchè non miglioriamo l'effetto della musica aggiungendo un paio di effetti visivi?

Gli effetti speciali

Il primo è molto semplice, si tratta di aggiungere una linea che cambia il colore dello sfondo e del bordo a seconda della nota letta:

```
870 POKE 53280,L:POKE 5328,H
```

Questa linea, come si può notare riguardando il programma di prima, modifica i colori dello schermo (53280 = bordo, 53281 = sfondo) a seconda della nota letta, (infatti utilizza gli stessi valori H e L) permettendoci di "vivacizzare" la musica.

Alla fine del ciclo però i colori risulteranno alterati, bisogna perciò rimetterlo a posto riPOKando i giusti valori nelle locazioni 53280/53281:

```
940 POKE 53280,2:POKE 53281,2
```

che rimette lo schermo tutto in rosso.

Ma siccome io sono un tipo che non si accontenta mai, e ho deciso di aggiungere un ulteriore effetto ottico nel programma, questa volta però in LM (Linguaggio Macchina), in modo da accontentare anche coloro che vogliano imparare questo linguaggio, difficile solo in apparenza.

L.M. per giochi

Cancelliamo la **linea 1100** e aggiungiamo le **linee 1120, 1130** e **1200** del listato in testa alla pagina.

EVM - COMPUTER

52025 MONTEVARCHI (Arezzo)
Via Marconi, 9/a
Tel. (055) 98.02.42 - 98.25.13

ESTRATTO DAL NS. CATALOGO GENERALE COMMODORE 64

GESTIONALI

CGD012 Contabilità ordinaria	230.000
CGD002 Magazzino (collegato)	150.000
CGD003 Fatturazione (collegato)	150.000
CGD018 Gestione C/C multipli	40.000
CGD011 Arredograph	195.000
CGD015 Gestione Biblioteca	80.000
CGD007 Gestione indirizzi	60.000
CUN053 Easy script nastro	50.000

UTILITY

CUD022 Pascal OXFORD	150.000
********** Assembler C/N/D	da 35.000
CUD029 Isam	75.000
CUD082 Simon's Compiler	70.000
CUD081 Compil. BLITZ	60.000
CUD038 Unguard	120.000
CUD039 Clone	80.000
CUD103 Bisector	80.000
CUD104 Pirate Disk	70.000
CUD053 Copy 190	70.000
CUD040 Turbo Disk	60.000

HARDWARE

CHV008 Programmatore EPROM *64	250.000
GVV004 Monitor 14" colore con audio orientabile CABEL	510.000
CHV006 Interfaccia CENTRONICS	95.000
GVD006 Copricomputer plastica	19.000

CHV021 Backup cassetta	35.000
GVD001 Dischi 3M/DATALIFE	3.950
GVW006 Nastri 801/802	15.000
GVD002 Kit pulizia disco (2)	20.000
GVD012 Ricambi per Kit (10)	30.000
GVD003 Contenitore 10 dischi	6.000
GVD004 Contenitore 40 dischi	29.000

MANUALI

CMB050 Guida al CBM64	28.000
CMB051 Sistema operativo CBM64 + SUPERMON	38.000
GMB054 Periferiche COMMODORE	25.000
GMB055 I segreti del 1541	28.000
CMB056 Corso di grafica CBM64**	24.000
CMB057 Corso di Assembler CBM64**	38.000

** Completi di programmi nastro

MANUALI PROGRAMMI

CMB008 Wedge 4.0	10.000
CMB014 Vizawrite	15.000
CMB017 Simon	20.000
CMB018 Master	25.000
CMB019 Extended basic	8.000
CMB030 Pet speed	15.000
CMB033 Easy script	20.000
CMB034 Tool	15.000
CMB035 Superbase	25.000
CMB036 Clone	10.000

CMB037 Unguard	10.000
CMB038 Pascal Oxford (inglese)	20.000
CMB039 Kmmm-Pascal	10.000
CMB040 Sam reciter	15.000
CMB041 Calc result	15.000
CMB042 Multiplan HELP	12.000
CMB046 Easy file	25.000
CMB060 Super expander	15.000

OFFERTE SPECIALI

Espansione 16K Vic 20 (1)	90.000
Espansione 32K Vic 20 (1)	120.000
Grafica + 3K VIC 20 (2)	54.000

(1) Con manuale Guida VIC 20
(2) Con manuale Perif. VIC 20

BIBLIOTECA 64 — 65.000
Comprende: Guida al CBM 64, Sistema Operativo CBM64, I segreti del 1541, Schema elettrico disco, SUPERMON, Schema elettrico CBM64.

Cognome ____________________

Nome ____________________

Indirizzo ____________________

CONDIZIONI DI VENDITA - La merce viene resa FRANCO Montevarchi. Per spedizioni in contrassegno calcolare L. 5.500 per rimborso spese postali e varie. Con pagamento anticipato non saranno addebitate. Il pagamento anticipato può essere effettuato anche con Assegno di c/c. Non inviare contanti o francobolli. I prezzi sono al netto di IVA del 18% mentre nei manuali è compresa IVA 2%. CATALOGO GRATUITO A RICHIESTA. Al primo ordine o alla richiesta di catalogo il Vs. nome sarà inserito nella EVM MAILING LIST e verrete tenuti periodicamente informati delle novità sul Vs. computer.

```
1110 S=0:FOR K=0 TO 13:READ A:S=S+A:POKE 833+K,A:NEXT
1120 READ A:IF A<>S THEN PRINT" ERRORE NEI DATI LM":END
1130 RETURN
1200
DATA169,0,170,168,238,32,208,232,208,250,200,208,247,96.
```

Queste linee, oltre a leggere i dati della routine LM ed a controllarne l'esattezza analogamente a quanto succedeva prima per i dati musicali, esegue anche un'altra operazione: memorizza (POKE) questi dati dalla locazione di memoria 833 in poi (vedasi POKE 833+K,A in **riga 110**).

Già, perchè se i dati musicali possono essere letti e "suonati" uno alla volta, un programma LM, per quanto breve che sia, deve essere posto da qualche parte nella memoria per essere eseguito.

E' come nel basic, voi non potete ad esempio caricare da cassetta o da disco un programma una linea alla volta ed eseguirla, ma dovete avere in memoria tutto il listato completo. Nel nostro caso, "qualche parte di memoria" si trova dall'indirizzo 826 all'indirizzo 1023. Questo pezzo di memoria serve normalmente per operazioni col registratore, (LOAD, SAVE, ecc.) ma siccome il gioco non le utilizza, per noi è un "buco" libero, dove possiamo metterci quello che vogliamo.

Se provate ora a dare RUN al programma così modificato noterete che... non è cambiato niente. Infatti, per eseguire una routine in LM non è solo necessario metterla da qualche parte, ma bisogna anche dire al computer che c'è e dove inizia, altrimenti per lui i dati di una routine in LM, di sprite, di musica, ecc. sarebbero numeri senza senso, che ignorerebbe.

Entrare in L.M

L'istruzione che dice al computer dove si trova una determinata routine in LM e fa passare il controllo a questa è SYS (indirizzo), che fa "partire" appunto la routine in LM che è allocata dalla locazione "indirizzo" (nel nostro caso, 833).

Cambiamo quindi la **linea 930** in:
930 SYS(833)
che ci permette di raggiungere lo scopo prefissato. Ora, se avete fatto tutto esattamente, fate qualche partita.

Quando raggiunge il bersaglio, dopo la musica noterete uno strano effetto sul bordo dello schermo, che si riempirà come di "righe colorate" per qualche secondo e poi tornerà ai colori originali.

A quanto abbiamo capito finora questa routine ha a che fare con il colore del bordo, ma vediamo di analizzarla meglio, ma non partendo dai DATA, ma dall'equivalente disassemblato della routine (il disasemblato di LM è l'equivalente del listato per i programmi basic).

Attenzione: Le linee del listato al piede della pagina non sono da digitare in quanto vengono poste in memoria automaticamente dai dati della **linea 1200** e dal ciclo delle **linee 1110-1130**.

In locazione (d'ora in poi la chiameremo linea) **833** c'è l'istruzione LDA≠00. Questa istruzione per il "cervello" al silicio 6510 significa: "carica nell'accumulatore il valore 0" (loaD Accumulator). L'accumulatore, così come i registri X e Y sono delle memorie che usa il 6510 per "ricordare" qualcosa. In basic le potremmo chiamare variabili. Questi possono contenere solo valori numerici, e questi non possono essere minori di 0 o maggiori di 255.

Le **linee 835 e 836** trasferiscono il valore che c'è in A nei registri X e Y (TAX= Transfer A in X, TAY=Transfer A in Y). Abbiamo così ottenuto l'azzeramento dei tre registri del microprocessore 6510:"A", "X" e "Y". La **linea 837** "prende" il contenuto della locazione specificata (in questo caso 53280), lo incrementa di + 1 e lo "risbatte" nella stessa locazione (INC = Increment memory).

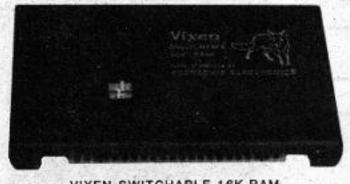
La **linea 840** incrementa il rtregistro X di uan unità per usarlo come contatore (ricordaiamo che all'inizio X era a 0).

La **linea 841** controlla (in questo caso) che il registro X non sia arrivato a 0. In caso sia diverso da 0 ritorna a **837**, dove continua a cambiare il colore fino al nuovo azzeramento di X.

A questo punto qualcuno potrebbe dire: "Abbiamo detto che X all'inizio era 0, poi, con istruzioni successive, continuiamo ad incrementarlo, ossia ad aumentarne il valore. Come è possibile allora che questo valore ritorni ad un certo punto ad essere 0?

Come detto in precedenza, A, X e Y non possono contenere un valore superiore a 255, per cui, se ad esempio X contiene 255 e lo incrementiamo di 1, non viene segnalato nessun errore, ma semplicemente, il valore, invece di diventare 256 come dovrebbe, ritorna a 0.

loc#	ist#	codici DATA
833	LDA #00	(169 0)
835	TAX	(170)
836	TAY	(168)
837	INC 53280	(238 32 208)
840	INX	(232)
841	BNE 837	(208 250)
843	INY	(200)
844	BNE 837	(208 247)
846	RTS	(96)

DAI MEMORIA AL TUO VIC-20

VIXEN SWITCHABLE 16K RAM

CON QUESTA CARTRIDGE POTRAI ESPANDERE LA MEMORIA DEL TUO VIC-20 COME VORRAI: 3K-8K-16K

Progettata e prodotta in Inghilterra dalla Stonechip Electronics questa cartuccia, tramite il semplice posizionamento di microswitch, ti consente un ampliamento differenziato della **RAM: 3K o 8K o 16K,** secondo quanto richiede il programma che devi far girare. I **contatti placcati in oro,** assicurano una lunga durata a questa cartridge che darà nuova vita al tuo VIC-20. Un prezzo introvabile e **garanzia per 6 mesi.**

PROGRAMMERS AID CARTRIDGE

Questa cartuccia aggiunge ben **19 comandi** al tuo **VIC-20:** renumber, autoline, edit, sound, cursor, block, delete .... e altri. E' un prodotto della Stonechip Electronics con **6 mesi di garanzia** integrale.

MODULO DI ESPANSIONE PER VIC-20

Questo modulo di espansione differenziata, prodotto dalla Stonechip Electronics, ti consente di **utilizzare contemporaneamente** (posizionando i microswitch di cui é dotato) da **2** a **4 cartridges.** Aggiunge inoltre alla **ROM 4K.** Inserendo con l'espansione Vixen 16K Ram una espansione standard di 8K si ottengono ben **24K RAM.** E' anche dotato di una presa per collegare un ulteriore modulo di espansione. Garanzia 6 mesi.

TRE CARTRIDGES COMMODORE PER VIC-20 AL PREZZO DI UNA

Scegli i tre videogiochi che preferisci a Lire 41.000 + IVA
Codice 201: Sargon chess - Omega race - Avenger
Codice 202: Omega race - Voodoo castle - Rat race
Codice 203: Ride Fort Knox - Avenger - Mole attack
Codice 204: Gorf - Omega race - Cosmic cruncher

COMPUTER CASSETTE C-20

Nastri magnetici da venti minuti ad altissima affidabilità, specifici per computer. Nastro PE 619 AGFA, scorrimento su perni in acciaio autolubrificanti, assemblate con cinque viti, corpo elegantissimo completamente trasparente, complete di etichette e cartoncino.

INTERFACCIA PER REGISTRATORE PER VIC-20 E PER CBM-64

La Stonechip Electronics é stata la prima ditta in Europa a realizzare l'interfaccia per utilizzare qualsiasi registratore con i computer Commodore. Continuamente aggiornata é di certo una delle più affidabili presenti sul mercato. Garanzia di 6 mesi.

LIGHT-PEN PER VIC-20 E CBM-64

La penna ottica é uno dei più collaudati prodotti della Stonechip Electronics. Ha un'ottima stabilità e buona risoluzione verticale/orizzontale. Completa di cassetta programmi. Un prezzo conveniente e una garanzia di 6 mesi.

50 FANTASTICI VIDEOGIOCHI 50 PER VIC-20 O CBM-64

Registrati su nastro, della software-house inglese Cascade Games, all'**inverosimile prezzo** di sole L. **25.000 + IVA.**

Spedire il presente **MODULO D'ORDINE,** o fotocopia, in una busta chiusa, unendo L. 2.600 in francobolli per spese postali.
UN OMAGGIO A CHI ORDINA ALMENO DUE ARTICOLI

Spett.le **APCO s.r.l. - Cas. Post. 239 - 10015 IVREA (To)**
desidero ricevere quanto da me contrassegnato con X. Pagheró direttamente al postino gli importi qui elencati, che sono comprensivi di IVA e di spese di imballo e contrassegno.

☐ Vixen Switchable 16K RAM	a L. 95.000
☐ Programmer Aid Cartridge	a L. 48.000
☐ Modulo di espansione per VIC-20	a L. 75.000
☐ 3 cartridges Commodore codice	a L. 50.000
☐ 12 computer cassette C-20	a L. 15.000
☐ Interfaccia per registratore	a L. 31.000
☐ Light-Pen per VIC-20 e CBM-64	a L. 65.000
☐ 50 videogiochi per ☐ VIC-20 ☐ CBM-64	a L. 30.000

☐ Non acquisto nulla ma desidero ricevere i Vostri cataloghi

Nome e Cognome ____________________

Via ____________________

Città ____________________ CAP. __________

data ____________________ firma ____________________

Questa operazione, però, altera il FLAG, che viene automaticamente controllato dall'istruzione BNE, alterando così il normale flusso del programma.

Abbiamo così ottenuto di cambiare 256 volte (da 0 a 255) il colore del bordo dello schermo, però, utilizzando il LM, questa operazione è talmente veloce da non risultare visibile o quasi dall'occhio dell'osservatore. Bisogna perciò trovare il modo di aumentare il numero di cambi di colore, in maniera che l'effetto si noti almeno per qualche secondo.

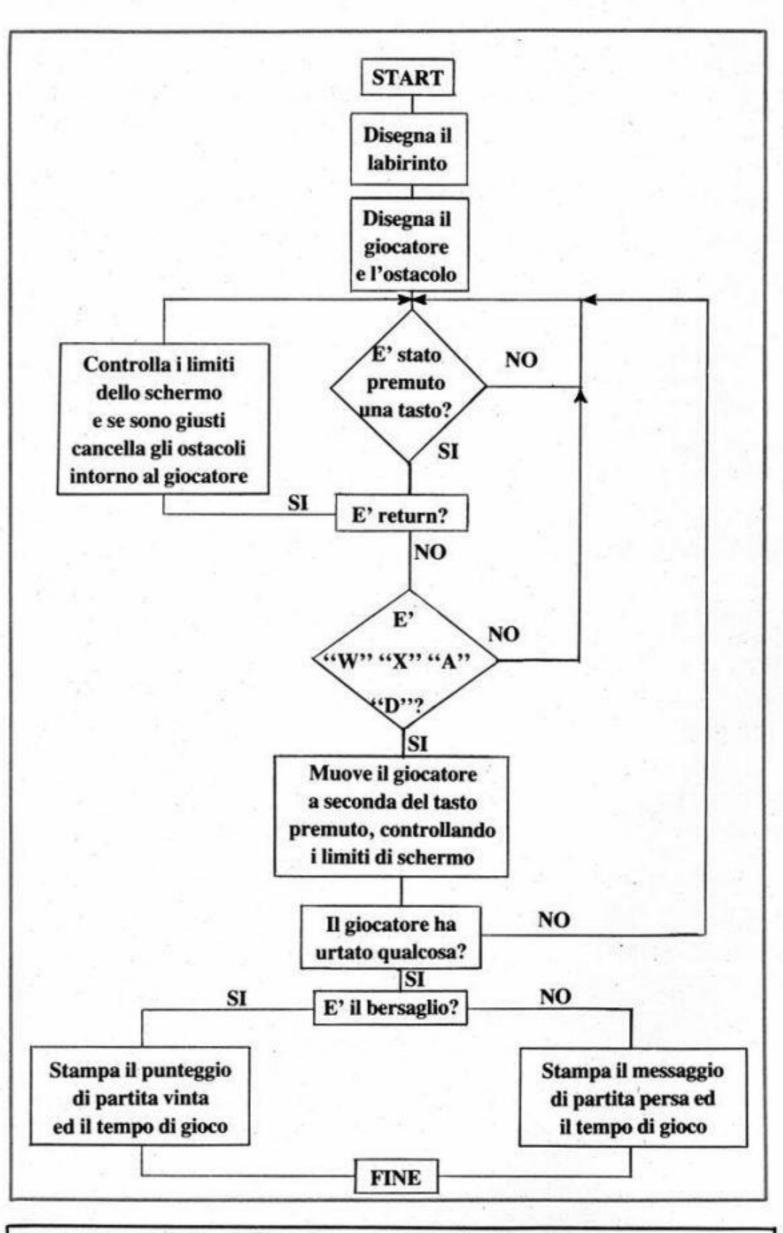
Siccome abbiamo detto che X e Y non possono contenere un valore superiore a 255, un solo registro non ci basta più come contatore, ma dovremo utilizzarne due o più. Abbiamo ancora il registro Y libero (anch'esso, ricordiamo, Azzerato all'inizio) e possiamo incrementarlo ogni volta che X ha finito 256 cicli.

A questo scopo serve la **linea 843.** La **linea 844** controlla con il solito BNE che Y non sia arrivato a 0, ritorna alla **linea 837**, dove si riinizia il ciclo di 256 cambi di colore con la X, e così via, per 256 volte, fino a che Y non raggiunge di nuovo il valore 0.

In questo modo, abbiamo ottenuto di cambiare il colore del bordo 256* 256=65536 volte! Si può notare, in questo caso, l'altissima velocità del LM che esegue tutto questo (in più o meno) due secondi.

E' appunto il cambio di colore così rapido (circa 30000 colori al secondo) che produce l'effetto di "strisce colorate" sullo schermo.

Ora però bisogna comunicare al 6510 che la routine è finita e che bisogna riprendere il programma basic. A questo scopo serve l'istruzione RTS (Return from Subroutine) della linea 846, che restituisce il controlo al basic.

Una precisazione da fare è che noi non possiamo mettere la routine in memoria così come è scritta, ma dobbiamo prima convertirla in numeri DATA e poi farla leggere dal calcolatore (vedasi linee 1110, 1130 e 1200)

Per chi ha il C16 o il Plus 4

Tutto quanto detto finora è valido anche per il C16/PLUS4 a patto di fare le seguenti modifiche:

- **Modificare la linea 160 in:**
  160 PRINT CHR$(147)
- **Cambiare la linea 180 in:**
  180 SC=3072
- **Cancellare la linea 190 e anche qualsiasi istruzione POKE che faccia riferimento al colore (nel programma sono le varie POKE CL+ ...eccetera.)**
- **Cambiare il PEEK(197) della linea 370 in PEEK(198)**
- **Cambiare i valori delle linee da 540 a 570:**
  1984 diventa 4032
  1064 diventa 3112
  2023 diventa 4071
  1025 diventa 3072

Tutto quanto si dirà invece nella seconda parte dell'articolo va bene solo per il CBM64.

```
100 GOSUB  1060:REM  LETTURA DATI
110 GOSUB  1000:REM  SCHERMATA INI
    ZIALE
120 POKE  198,0:REM  AZZERA BUFFER
    DI TASTIERA
130 :
140 REM  ** LISTATO 1 **
150 REM  *** DISEGNA LABIRINTO CAS
    UALE ***
160 POKE  53280,2:POKE  53281,2:PR
    INT CHR$(158):REM  CAMBIA I CO
    LORI
170 PRINT CHR$(147):REM  CANCELLA
    SCHERMO
180 SC=1024:REM  MEMORIA DI SCHERMO
190 CL=55296:REM  MEMORIA COLORE
200 INPUT "OSTACOLI (30-500)";N:RE
    M  CHIEDE IL NUMERO DI OSTACOL
    I DA STAMPARE
210 PRINT CHR$(147)
220 FOR  K=1  TO  N
230 R=INT(RND(1)*1000)+1:REM  GENE
    RA UN NUMERO CASUALE TRA 1 E 1
    000
240 POKE  SC+R,160:POKE CL+R,1:REM
     STAMPA GLI OSTACOLI IN BIANC
    O
250 NEXT K:REM  FINE CICLO
260 :
270 REM  ** LISTATO 2 **
280 REM  *** STAMPA GIOCATORE E BE
    RSAGLIO ***
290 POKE  SC,81:POKE  CL,0:REM  ST
    AMPA IL BERSAGLIO IN NERO
300 POKE  SC+999,87:POKE  CL+999,5
    :REM  STAMPA IL GIOCATORE IN V
    ERDE
310 :
320 REM  ** LISTATO 3 **
330 REM  ** INIZIALIZZAZIONE DELLE
     VARIABILI **
340 TI$="000000":REM  AZZERA IL TE
    MPO
350 R=24:C=39:REM  RIGA E COLONNA
    INIZIALI DEL GIOCATORE
360 REM  *** INIZIO GIOCO ***
370 P=PEEK(197):IF  P=64  THEN  37
    0:REM  ATTESA DELLA PRESSIONE
    DI UN TASTO
380 IF  P=1  THEN  GOSUB  510:REM
     CONTOLLO PER PRESSIONE DI 'RE
    TURN'
390 POKE  SC+R*40+C,32:REM  CANCEL
    LAZIONE GIOCATORE DALLA POSIZI
    ONE PRECEDENTE
400 C=C+((P=10)-(P=18)):R=R+((P=9)
    -(P=23)):REM  AGGIORN. R/C A S
    ECONDA DEL TASTO
410 REM  * CONTROLLO LIMITI *
420 IF  C<0  THEN  C=0
430 IF  R<0  THEN  R=0
440 IF  R>24  THEN  R=24
450 IF  C>39  THEN  C=39
460 REM  * CONTROLLA SE IL GIOCATO
    RE VA CONTRO A QUALCOSA *
470 IF  PEEK(SC+R*40+C)<>32  THEN
     P=PEEK(SC+R*40+C):GOTO  610
480 POKE  SC+R*40+C,87:POKE  CL+R*
    40+C,5:REM  STAMPA GIOCATORE N
    ELLA NUOVA POSIZ
490 GOTO  370:REM  CONTINUO GIOCO
500 :
510 REM  *** TASTO RETURN PREMUTO:
    CANCELLAZIONE DI OSTACOLI VICI
    NI ***
520 S=SC+R*40+C:REM  MEMORIZZAZION
    E POSIZIONE GIOCATORE IN S
530 REM  ** CONTROLLO LIMITI PER C
    ANCELLARE **
540 IF  S<1984  THEN  POKE  S+40,3
    2
550 IF  S>1064  THEN  POKE  S-40,3
    2
560 IF  S<2023  THEN  POKE  S+1,32
570 IF  S>1025  THEN  POKE  S-1,32
580 FOR  K=1  TO  1000:NEXT:REM  C
    ICLO DI ATTESA
590 RETURN:REM  RITORNO AL PROGRAM
    MA PRINCIPALE
600 :
610 REM  ** LISTATO 4 **
620 TM$=TI$:REM  MEMORIZZAZIONE TE
    MPO
630 REM  *** FINE GIOCO ***
640 IF  P=81  THEN  790:REM  CONTR
    OLLA SE SI E' ANDATI CONTRO AL
     BERSAGLIO
650 REM  *** CONTRO AD UN OSTACOLO
     ***
660 FOR  I=54272  TO  54296:POKE
    I,0:NEXT:REM  AZZERAMENTO REGI
    STRI MUSICALI
670 REM  *** RUMORE ESPLOSIONE ***
680 POKE  54296,10:POKE  54272,214
```

```
    :POKE  54273,28:POKE  54277,11
690 POKE  54278,15:POKE  54276,129
700 REM  *** CAMBIA I COLORI DEL B
    ORDO ***
710 FOR  K=1  TO  15:POKE  53280,K
    :FOR  K1=1  TO  100:NEXT K1,K:
    POKE  53280,2
720 REM  *** SCHERMATA FINALE ***
730 PRINT CHR$(147)
740 PRINT" DISTRUTTO!!"
750 PRINT" TEMPO:";TM$
760 FOR  K=54272  TO  54296:POKE
    K,0:NEXT
770 POKE  198,0:PRINT" ...CIAO":EN
    D :REM  AZZERA BUFFER DI TASTI
    ERA E FINISCE
780 :
790 REM  ** RAGGIUNTO BERSAGLIO **
800 RESTORE :REM  AZZERA PUNTATORE
     DI LETTURA DATI
810 FOR  K=54272  TO  54296:POKE
    K,0:NEXT
820 REM  *** INIZIALIZZAZIONE VARI
    ABILI PER LA MUSICA ***
830 V=54296:W=54276:A=54277:HF=542
    73:LF=54272:S=54278:PH=54275:P
    L=54274
840 POKE  V,10:POKE  W,65:POKE  A,
    190:POKE  PH,15:POKE  PL,15:RE
    M  VALORI INIZIALI
850 REM  *** LETTURA DATI MUSICALI
     E CONTROLLO PER LA FINE ***
860 READ  H,L,D:IF  H=-1  THEN  FO
    R  K=54272  TO  54296:POKE  K,
    0:NEXT:GOTO  920
870 POKE  53280,L:POKE  53281,H:RE
    M  CAMBIO DI COLORE A SECONDA
    DELLE NOTE
880 REM  *** PONE LE NOTE E LA DUR
    ATA NEI GIUSTI REGISTRI ***
890 POKE  HF,H:POKE  LF,L:FOR  X=D
    -50  TO  D-20:POKE  S,136:NEXT
900 FOR  T=1  TO  D:NEXT:REM  DURA
    TA DELLE NOTE
910 POKE  HF,0:POKE  LF,0:POKE  W,
    0:GOTO  840:REM  AZZERAMENTO N
    OTE E CONTINUO
920 REM  *** FINE MUSICA (SCHERMAT
    A FINALE) ***
930 SYS 833:REM  RICHIAMO DELLA RO
    UTINE IN LINGUAGGIO MACCHINA
940 POKE  53280,2:POKE  53281,2:RE
    M  RIMESSA A POSTO DEI COLORI
    SFONDO/BORDO
950 PRINT CHR$(147)
960 PRINT" BRAVISSIMO, HAI RAGGIUN
    TO IL BERSAGLIO"
970 PRINT" TEMPO:"TM$
980 PRINT:PRINT" ...CIAO":POKE  19
    8,0:END :REM  AZZERA BUFFER DI
     TASTIERA E FINISCE
990 :
1000 REM  *** SCHERMATA INIZIALE **
     *
1010 PRINT CHR$(147):POKE  53280,2:
     POKE  53281,2:PRINT CHR$(158)
1020 PRINT" RACE-MAZE BY MARIANI G.
     "
1030 PRINT" PREMI UN TASTO PER INIZ
     IARE"
1040 POKE  198,0:WAIT  198,1:RETURN
     :REM  ATTESA PER LA PRESSIONE
     DI UN TASTO
1050 :
1060 REM  ** LETTURA DATI **
1070 S=0:FOR  K=1  TO  39:READ  A:S
     =S+A:NEXT:REM  DATI MUSICALI
1080 REM  ** CONTROLLO PER DATI MUS
     ICALI **
1090 READ  A:IF  S<>A  THEN  PRINT"
     ERRORE NEI DATI MUSICALI!":END

1100 REM  *** LETTURA DATI LM ***
1110 S=0:FOR  K=1  TO  14:READ  A:S
     =S+A:POKE  832+K,A:NEXT
1120 READ  A:IF  A<>S  THEN  PRINT"
     ERRORE NEI DATI LM":END :REM
     CONTROLLO PER DATI LM
1130 RETURN
1140 :
1150 REM  ** DATI PER LA MUSICA **
1160 DATA  34,75,250,43,52,250,51,9
     7,375,43,52,125,51,97,250,57,1
     72,250,51,97
1170 DATA  500,0,0,125,43,52,250,51
     ,97,250,57,172,1000,51,97,1000
     ,-1,-1,-1
1180 DATA  6214:REM  CHECKSUM DATI
     MUSICA
1190 REM  ** DATI LM:ATTENZIONE A C
     OPIARLI ESATTAMENTE **
1200 DATA  169,0,170,168,238,32,208
     ,232,208,250,200,208,247,96,24
     26
```

CARATTERI E CODICI DEL C 64

Simbolo	Tasti utilizzati	Denominaz.	Simbolo	Tasti utilizzati	Denominaz.	Simbolo	Tasti utilizzati	Denominaz.	Simbolo	Tasti utilizzati	Denominaz.
	Shift Clr/Home	Clear		Ctrl 1	Nero		Commodore 1	Aranc		F1	F1
	Clr/Home	Home		Ctrl 2	Bianco		Commodore 2	Marr		Shift F1	F2
	Shift Crsr (S)	Up		Ctrl 3	Rosso		Commodore 3	Rosa		F3	F3
	Crsr (S)	Down		Ctrl 4	Azzur		Commodore 4	Grigio 1		Shift F3	F4
	Shift Crsr (D)	Left		Ctrl 5	Viola		Commodore 5	Grigio 2		F5	F5
	Crsr (D)	Right		Ctrl 6	Verde		Commodore 6	Verde 2		Shift F5	F6
	Ctrl 9	Rvs		Ctrl 7	Bleu		Commodore 7	Celeste		F7	F7
	Ctrl 0	Rvoff		Ctrl 8	Giallo		Commodore 8	Grigio 3		Shift F7	F8

Calcio Computer, *8 programmi software per Commodore 64, che di volta in volta ti insegneranno a compilare la schedina (e possibilmente vincere), elaborare i tuoi pronostici sulla classifica di campionato, raccogliere e consultare i dati delle squadre e dei giocatori di serie A e B.*
Con ***Calcio Computer*** *si inaugura* ***Mondosoft,*** *il nuovo periodico per home computer della Mondadori.*
Ogni quattordici giorni in edicola con un nuovo programma su cassetta corredato da un fascicolo a colori.
Mondosoft, *per esplorare i vari "mondi" del software.*

MONDOSOFT

Il periodico che leggi con il computer.

COMPUTER COMMODORE e DRIVE 1541

GESTIONE DI ARCHIVI SU DISCO

Imparare a creare archivi mediante l'uso di semplici routines che gestiscono files su disco.

Il lettore interessato alla manipolazione dei dati su disco può trovare nelle pagine che seguono un utile riferimento nella programmazione dell'INPUT-OUTPUT. In questa prima parte vengono esposte quattro routines per utilizzare files sequenziali (creazione, lettura, scrittura, controllo errori). Viene inoltre proposto un programma che utilizza le routines per la soddisfazione dei lettori interessati alla sola digitazione e come esempio didattico per chi si propone di approfondire l'argomento.

Cos'è un file?

Tutte le informazioni presenti nella memoria RAM - istruzioni (programmi) e dati - vengono inevitabilmente perdute se togliamo l'alimentazione o procediamo a nuove memorizzazioni. Le memorie di massa (nastri, dischi), permettono di ovviare a questo inconveniente consentendoci di registrare o leggere le informazioni sotto forma di files.

I files sono quindi strutture di istruzioni o di dati; può essere utile, come chiarimento, pensare al dischetto come ad una libreria dove ogni libro rappresenta un file. La gestione dei files di istruzioni, detti files-programma, è affidata ai comandi SAVE, LOAD, VERIFY di cui tutti i lettori hanno buona conoscenza. Vengono invece qui trattati i files dati, che possono avere differente struttura in relazione al modo con cui si intende trattarne i contenuti.

Un Data-file è costituito da un insieme di unità di informazioni omogenee dette record; ogni record a sua volta è diviso in campi. Per esempio, il file contenente l'elenco degli studenti di una certa classe potrebbe essere costituita da record come in figura 1. Il file conterrà tanti record quanti sono gli studenti: la struttura interna del record viene quindi definita dall'utente.

I files su disco nel Commodore 64 possono essere organizzati in due modi diversi: ad accesso sequenziale, ad accesso casuale.

I file sequenziali sono strutturati come le canzoni su un normale nastro da registratore, i record vengono scritti uno dopo l'altro; per rileggere l'ultimo occorre scandire il file dall'inizio alla fine. Una eventuale espansione deve essere effettuata caricando l'intero file in memoria per poi registrarlo nuovamente con gli eventuali aggiornamenti o aggiunte. La caratteristica più macroscopica è comunque il fatto che non si può accedere all'i-esimo record in modo diretto e veloce.

I files ad accesso casuale permettono invece di reperire in ogni istante il record

voluto senza scandire l'intero archivio (file-dati). Nel drive 1541 tali files prendono il nome di RELATIVE e presentano l'unico inconveniente dell'obbligo di assegnare (e rispettare rigorosamente in seguito) una lunghezza precisa al record così come ai singoli campi che lo compongono. In modo da poter accedere ad un determinato record o ad un determinato campo con estrema precisione e velocità.

Routine di inizializzazione archivi

Il primo passo da compiere per lavorare con i files dati è la loro creazione; questo è infatti lo scopo della prima routine proposta. Prima di affrontare le righe di programma è opportuno però premettere che esiste una procedura da rispettare nella trattazione dei files, detta "apertura del canale".

L'istruzione:
OPEN nf,np,nc,"0:nome file,tf,df"
permette di espletare tale procedura.

Per numero di file (nf), si intende un numero da 1 a 255 che servirà per identificare il file nelle istruzioni di input-output; il numero di periferica (np), è normalmente 8 se si possiede un solo drive; il numero di canale (nc), identifica la via di comunicazione con il disco. Si tenga presente comunque che i canali 1 e 2 sono usati nelle operazioni di LOAD e SAVE dal sistema operativo ed il canale 15 viene usato come canale di comando, su cui torneremo in seguito.

Nelle virgolette oltre alla formula fissa 0: (facoltativa per chi possiede un solo 1541), si deve indicare il nome del file, il tipo (tf) che risulta 's' per sequenziale e la direzione (df): 'w' (=write= scrittura),oppure 'r' (=read =lettura).

L'istruzione:
OPEN 1,8,3,"0:archivio,s,w"
aprirà quindi un file sequenziale in scrittura su disco con il nome "archivio", sul canale 3 identificabile con il numero 1. Una volta eseguita questa proced ura il file potrà essere scritto (o letto), semplicemente usando le istruzioni PRINT nf, (INPUT nf, GET nf), dove nf è appunto l'identificatore del file su cui vogliamo lavorare.

Ritornando alla creazione degli archivi, l'esame della routine che inizia alla riga 10000 pone subito in evidenza la procedura utilizzata per l'apertura del file da creare. Tutti i parametri sono, i nfatti, dichiarati tramite la stringa NF$; questo sistema permette di aprire ogni volta un file con nome diverso prelevato in input. La routine 15000, chiamata in causa subito dopo, provvede al controllo di eventuali errori ed il flag 'f' permette di recuperarli.

Nelle righe 10100 - 10110, si provvede alla scrittura delle prime informazioni. E' necessario notare che ogni print effettua la scrittura di una singola unità di informazione. All'interno di un file sequenziale, i dati devono essere separati l'uno dall'altro mediante caratteri speciali. Nonostante il manuale del 1541 indichi come separatori validi il tasto return,la virgola e il punto e virgola, nella pratica gli ultimi due generano molteplici problemi. In fase di lettura, infatti, quasi sempre tali caratteri non vengono riconosciuti generando così l'accumulo di più dati in una sola variabile e ne consegue (con l'istruzione INPUT) che quando si superano i 256 caratteri viene emesso l'errore "String too long".

Risulta evidente che l'unico separatore valido è il carattere chr$(13) che deve assolutamente essere frapposto almeno ogni 80 caratteri in modo da evitare, in fase di INPUT# , l'errore menzionato. E' ovvio che se si ricorre alla sola istruzione GET (in fase di successiva lettura) tale accorgimento può essere evitato.

Un modo veloce di utilizzare il cafrattere di ritorno carrello (13) consiste nel far seguire una sola variabile ad ogni istruzione PRINT

Il primo dato registrato (NR$ riga 12070) corrisponde al numero di record presenti nel file, ovviamente zero in fase di inizializzazione.

Il secondo dato (NC$) invece, corrisponde al numero dei campi dichiarati in input; indispensabile per definire il record in fase di lettura.

Il ciclo aperto in 10120, preleva in input i nomi dei campi per poi registrare gli stessi sul file. In questo modo l'archivio è inizializzato e contiene anche le indicazioni circa la struttura dei record.

In altre parole, per come è strutturato il programma pubblicato, i primi dati del file registrato su disco, avranno lo specifico compito di contenere le varie informazioni che interessano l'intero file.

Routine di caricamento archivio

Quando si lavora con gli archivi sequenziali, come già precedentemente accennato, è necessario caricare tutto l'archivio in memoria per apportare modifiche o aggiunte.

La routine che inizia in 11000 espleta questa funzione. Essa provvede all'apertura dell'archivio prelevandone il nome in input.

Nella riga 11080 vengono poi letti il n.record ed il n.campi, i cui valori indicano la quantità di dati da leggere; con questi si definisce infatti il range dei cicli necesssari a caricare una matrice stringa di due dimensioni (MD$), dove la riga 0 conterrà i nomi dei campi del record. La matrice in questione può essere dimensionata sia con i dati contenuti in NR$(righe)

e in NC$(colonne) che in un modo standard per poter trattare più file di diverse dimensioni nella stessa sessione.

E' necessario tenere presente, inoltre, che i file non devono superare la capacità di memoria utente disponibile per poter essere manipolati. Ad esempio, se il vostro programma occupa 10000 bytes, rimangono disponibili per i dati circa 29K (Commodore 64) questo significa che il vostro sequenziale non può superare tale grandezza per poter essere gestito completamente.

Routine di registrazione archivio

A lavoro ultimato, il file trattato deve essere registrato nuovamente pena l'utilità delle modifiche eseguite. La routine in riga 12000 sovraintende alla registrazione.

L'apertura del file, in questo caso, tiene conto della sua precedente esistenza su disco: l'opzione è indispensabile, infatti, per "ricollocare il file su se stesso". Tale comando, come si può notare, è "coperto" da REM dato che, usato spesso, provoca malfunzionamenti sul dischetto. Ecco perchè il file precedente viene cancellato (comando "scratch" di riga 12040) per essere nuovamente riscritto dopo l'aggiornamento.

Dopo l'apertura del canale vengono immediatamente scritti i contenuti di NR$, e NC$; il primo dovrà contenere il numero di record presenti nella matrice, comprensivo delle aggiunte o variazioni effettuate.

In seguito si procede alla scrittura dei dati nello stesso modo in cui si procedeva alla lettura, semplicemente sostituendo l'istruzione INPUT# con PRINT#.

Routine di lettura errore

Il drive 1541 comunica al sistema quattro dati, leggibili in qualsiasi momento; questi dati sono, nell'ordine, il codice di errore, il nome, la traccia e il blocco in cui l'errore è occorso. Un pro gramma che gestisce files deve necessariamente provvedere al controllo degli errori, onde evitare "impiantamenti" non desiderati.

Anche in questo caso è indispensabile l'apertura del canale con l'istruzione OPEN nf,8,15. Come si può notare il canale in questione è il n. 15 mentre il n.file può essere scelto liberamente tra quelli non utilizzati in altre aperture. Ogni volta che si inizia un colloquio con la periferica (open), si dovrà leggere e gestire l'eventuale errore. Si deve tenere presente, inoltre, che il canale 15 deve essere aperto per primo e chiuso per ultimo, evitando cosi rischiose perdite di dati.

La riga 15030 della routine proposta per questo lavoro, legge i quattro valori di cui sopra; la riga seguente ritorna il controllo alla routine di chiamata se non viene rilevato nessun errore. L'analisi viene fatta sul codice di errore; scorrendo la lista degli errori sulla guida del 1541 ci si rende subito conto che sono reali errori solo quelli corrispondenti ai codici da 20 a 74. In presenza di uno di questi, viene infatti, stampato un messaggio di errore e si pone uguale ad uno il valore del flag 'f' per segnalare l'errore alla routine di chiamata che in tal caso ritorna al main program senza perdere i dati eventualmente contenuti nella matrice MD$.

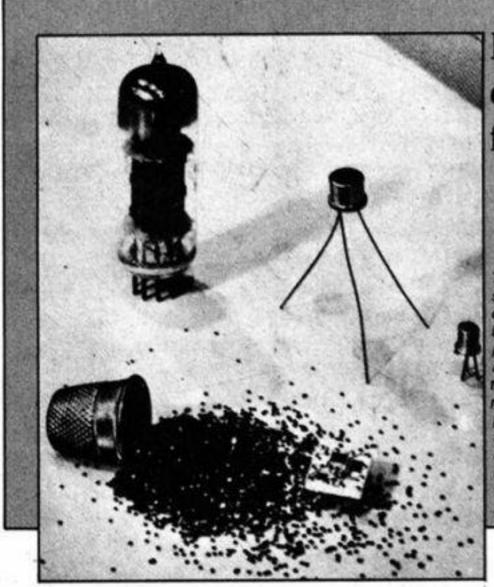
Il programma

Come già accenato nella premessa, il programma che gestisce le quattro routines può essere considerato sia come strumento di lavoro sui file, che come piattaforma di lancio per la preparazione di un software personale. Per questa ragione non viene descritto il suo funzionamento, il lettore interessato troverà, infatti, il listato stesso un valido strumento di studio. Chi desidera risultati immediati può mettere in funzione il tutto semplicemente trascrivendo il programma seguito dalle routines.

L'utility permette di creare e modificare archivi strutturati in max 100 record di max 10 campi.

Il menù che appare dopo il "RUN", visualizza tutte le possibilità del programma; a questo punto basta premere il numero desiderato ed il gioco è fatto.

Per sollevare il lettore da troppo lunghe digitazioni, il programma presentato appare "nudo" dal punto di vista grafico; una serie di "pennellate" alla facciata potranno renderlo più chiaro nella esecuzione; inoltre possono essere inseriti dei controlli negli input per evitare errori di inserimento così come sarebbe opportuno aggiungere al menù la voce "ordinamento" per chi desidera l'ordine alfabetico dei record introdotti.

D. Matturro M.L. Nitti

Le immagini di questo fascicolo

Tre generazioni a confronto: sullo sfondo, un tubo a vuoto caratteristico della cosiddetta 1 generazione di elaboratori elettronici e due transistor impiegati nella 2 seconda generazione; in primo piano, i transistor microminiaturizzati grandi come grani di sale raffinato - un ditale ne contiene 50.000 - presenti nell'unità centrale del Sistema†360. L'adozione dei microcircuiti integrati, che "scattano", in pochi miliardesimi di secondo, consente velocità operative mille volte superiori a quelle delle macchine precedenti.

ARCHIVIO SEQUENZIALE

```
100 REM  COMPUTER COMMODORE
110 REM     & DRIVE 1541
130 REM
140 REM   GESTIONE ARCHIVI
170 REM
180 REM        M.L.NITTI
185 REM      D. MATTURRO
190 REM
200 REM  ******************
210 :
220 DIM  MD$(100,10): REM  MATRICE
     DATI
230 OPEN  15,8,15: REM  APRE CANAL
    E DI ERRORE
240 GOSUB  15000: REM   LETTURA CA
    NALE DI ERRORE
250 PRINT CHR$(147): REM  PULIZIA
    SCHERMO
260 PRINT:PRINT TAB(10)"GESTIONE A
    RCHIVI"
270 PRINT:PRINT:PRINT:PRINT TAB(5)
    "[1] FINE"
280 PRINT TAB(5)"[2] CREAZIONE":PR
    INT TAB(5)"[3] CARICAMENTO"
290 PRINT TAB(5)"[4] REGISTRAZIONE
    ":PRINT TAB(5)"[5] LETTURA"
300 PRINT TAB(5)"[6] SCRITTURA":PR
    INT TAB(5)"[7] VARIAZIONE"
310 GET  A$: IF  A$="" GOTO 310
320 ON  VAL(A$) GOSUB  15110,5000,
    5000,12000,2000,1000,3000
330 GOTO  250
340 :
1000 REM      SCRITTURA ARCHIVIO
1005 :
1010 NR=VAL(NR$)+1: REM  NR= N.RECO
     RD GIA SCRITTI
1020 PRINTCHR$(147):PRINT TAB(10)"S
     CRITTURA ARCHIVIO"
1030 PRINT:PRINT"RECORD N.";NR:FOR
     K=1 TO VAL(NC$)
1040 PRINTMD$(0,K);"=";:INPUT  MD$(
     NR,K): REM  SCRITTURA CAMPO
1050 NEXT K:PRINT:PRINT TAB(5)"S PE
     R CONTINUARE"
1060 GET  A$: IF  A$=""GOTO 1060
1070 IF  A$<>"S" THEN  NR$=STR$(NR)
     :RETURN
1080 NR=NR+1:GOTO  1030 : REM  INCR
     EMENTA N.RECORD GIA SCRITTI
1090 :
2000 REM      LETTURA ARCHIVIO
2005 :
2010 IF VAL(NR$)<>0GOTO 2040 :REM
     ANALISI PRESENZA DATI
2020 PRINT:PRINT:PRINT TAB(9)"ARCHI
     VIO VUOTO"
2030 FOR K=1 TO 1500:NEXTK:RETURN
2040 PRINT CHR$(147):PRINT TAB(10)"
     LETTURA ARCHIVIO"
2050 FOR K=1 TO VAL(NR$): REM  CICL
     O LETTURA RECORD
2060 PRINT:PRINT"RECORD N.";K
2070 FOR  KK=1 TO VAL(NC$): REM  CI
     CLO LETTURA CAMPI
2080 PRINTMD$(0,KK);"=";MD$(K,KK)
2090 NEXTKK
2100 GET  A$:IF A$=""GOTO 2100
2110 NEXTK:RETURN
2120 :
3000 REM  VARIAZIONE ARCHIVIO
3005 :
3010 PRINT CHR$(147):PRINT TAB(10)"
     VARIAZIONE ARCHIVIO":PRINT
3020 NR=VAL(NR$): REM  N.RECORD SCR
     ITTI
3030 INPUT "N. RECORD";N
3040 IF N>NR+1  OR  N=0  THEN  PRIN
     TCHR$(145) CHR$(145): GOTO 303
     0
3050 PRINT:FOR K=1 TO VAL(NC$):REM
      CICLO DI INSERIMENTO AGGIORNA
     MENTI
3060 PRINTMD$(0,K);"=";:P=POS(0):PR
     INTSPC(2)MD$(N,K)
3070 PRINT CHR$(145)  TAB(P):INPUT
      MD$(N,K)
3080 NEXT K:IF  N=NR+1 THEN  NR$=ST
     R$(N): REM  INCREMENTA NR$ SE
     NECESSARIO
3090 PRINT:PRINT TAB(5)" S PER CONT
     INUARE"
3100 GET  A$:IF A$=""GOTO 3100
3110 IF A$<>"S" THEN  RETURN
3120 GOTO  3000
5000 REM  CANCELLAZIONE MATRICE DAT
     I
5010 FOR K=0 TO VAL(NR$):FOR KK=1 T
     O VAL(NC$):MD$(K,KK)="":NEXTKK
     ,K
```

```
 5020 NR$="0":NC$="0"
 5030 ON VAL(A$) GOSUB  ,10000,11000
      :RETURN
 5040 :
 5050 :
10000 :
10005 REM  INIZIALIZZAZIONE
10010 REM  ARCHIVIO SEQUENZIALE
10020 :
10030 PRINT CHR$(147):PRINT: F=0: RE
      M  F=FLAG DI CONDIZIONE DI ERR
      ORE
10040 INPUT "NOME ARCHIVIO";NA$: REM
       MAX 16 CARATTERI
10050 INPUT "NUMERO CAMPI";NC$
10060 NF$=NA$+",S,W": REM  STRINGA S
      PECIFICHE FILE
10070 OPEN  3,8,3,NF$ : REM  APERTUR
      A FILE ARCHIVIO
10080 GOSUB  15000 :REM  LETTURA CAN
      ALE DI ERRORE
10090 IF  F=1 THEN  CLOSE  3:RETURN:
      REM  USCITA PER RECUPERO ERROR
      E DISCO
10100 PRINT#3,0 : REM  REGISTRA NUME
      RO RECORDS
10110 PRINT#3,NC$: REM  REGISTRA NUM
      ERO CAMPI
10120 FOR  H=1  TO VAL(NC$): REM  CI
      CLO DI INSERIMENTO E REGISTRAZ
      IONE NOME CAMPI
10130 PRINT"NOME CAMPO ";H;: INPUTN$
10140 PRINT#3,N$ :REM  REGISTRA NOME
       CAMPO
10150 MD$(0,H)=N$
10160 NEXT H:CLOSE  3:RETURN: REM  C
      HIUDE FILE ARCHIVIO
11000 :
11010 REM  CARICAMENTO ARCHIVIO
11020 :
11030 PRINT:PRINT TAB(5)"NOME ARCHIV
      IO";
11040 F=0:INPUT  NA$
11050 NF$=NA$+",S,R": REM  STRINGA S
      PECIFICHE FILE RICHIESTO
11060 OPEN  3,8,3,NA$: REM  APERTURA
       FILE RICHIESTO
11070 GOSUB  15000:IF F=1 THEN CLOSE
        3:RETURN: REM  CONTROLLLO ER
      RORE
11080 INPUT#3,NR$,NC$: REM  LETTURA
      N.RECORD,N.CAMPI
11090 FOR K=0 TO VAL(NR$)
11100 FOR KK=1 TO  VAL(NC$)
11110 INPUT#3,MD$(K,KK): REM  LETTUR
      A FILE
11120 NEXTKK,K:CLOSE  3: RETURN
12000 :
12010 REM  REGISTRAZIONE ARCHIVIO
12020 :
12030 PRINT:PRINT:PRINT TAB(5)"REGIS
      TRAZIONE IN CORSO"
12035 REM  NF$="@0:"+NA$+",S,W": REM
        STRINGA SPECIFICHE FILE
12040 NF$="S0:"+NA$:PRINT#15,NF$: RE
      M  CANCELLA FILE PRECEDENTE
12045 NF$=NA$+",S,W": REM  STRINGA S
      PECIFICHE FILE
12050 OPEN  4,8,4,NF$:REM  APERTURA
      FILE
12060 GOSUB  15000:IF F=1 THEN CLOSE
        4:RETURN: REM  CONTROLLO ERR
      ORE
12070 PRINT#4,NR$: REM  SCRIVE N.REC
      ORD PRESENTI
12080 PRINT#4,NC$:REM  SCRIVE N.CAMP
      I PER RECORD
12090 FOR K=0 TO VAL(NR$)
12100 FOR KK=1 TO  VAL(NC$)
12110 PRINT#4,MD$(K,KK): REM  SCRITT
      URA DATI
12120 NEXTKK,K:CLOSE  4: RETURN
15000 :
15010 REM  LETTURA CANALE DI ERRORE
15020 :
15030 INPUT#15,E,E$,E1,E2: REM  LETT
      URA CODICI ERRORE
15040 IF E<20 OR E>74 THEN  RETURN:
      REM  CONTROLLO PRESENZA ERRORE
15050 PRINT:PRINT TAB(5)"ERRORE DISC
      O"
15060 PRINT TAB(4)E,E$:REM  STAMPA M
      ESSAGGIO ERRORE
15070 PRINT TAB(5)"TRACCIA=";E1;" BL
      OCCO=";E2: REM  STAMPA TRACCIA
       E BLOCCO
15080 PRINT:PRINT TAB(5)"S PER CONTI
      NUARE"
15090 GET  A$:IF A$="" THEN 15090
15100 IF A$="S" THEN  F=1:RETURN: RE
      M  F= FLAG ERRORE
15110 CLOSE  15:END
```

ADDENTA QUELLE RUOTE

Dimensionare ruote dentate cilindriche secondo norme UNI.

Il dimensionamento delle ruote dentate è un problema classico della meccanica che, pur non presentando grosse difficoltà sotto l'aspetto teorico, per contro richiede un notevole numero di calcoli con relative consultazioni di tabelle. Ovviamente questo lavoro è svolto dal programma che pubblichiamo, e tutto ciò che è richiesto è di conoscere alcune definizioni che diamo di seguito.

- **Diametro di testa:** è il diametro esterno della ruota dentata (misurato cioè sulla "testa" del dente).
- **Diametro di base:** (o interno) è, di contro, quello misurato sul "fondo" del dente.
- **Diametro primitivo:** (Dp) è il diametro a circa metà dentatura.

E' questo il più importante dei tre e del quale si tiene conto nel calcolo degli ingranaggi. Non vale esattamente la media tra i diametri interno ed esterno perchè i denti vengono dimensionati in modo tale che quando ingranano fra loro vi sia un po' di gioco, necessario per evitare rumorosità ed una rapida usura dei denti stessi.

A tal scopo questi ultimi vengono costruiti seguendo una forma (profilo) che permetta loro di lavorare nelle migliori condizioni.

Affinchè ciò avvenga si deve assolutamente evitare che:

- La testa del dente venga in contatto con la base dei denti dell'altra ruota. A questo si rimedia semplicemente costruendo il vano maggiorato rispetto alla testa (vedi figura 2).
- I denti nel loro movimento relativo striscino fra di loro.

Figura 1
Figura 2

E' evidente che se ciò si verificasse l'ingranaggio risulterebbe notevolmente rumoroso, senza contare che l'attrito generato dal contatto radente porterebbe ad un rapido deterioramento delle dentature.

Per far fronte all'inconveniente si profilano i denti seguendo una curva particolare detta "evolvente" (questa si ottiene facendo rotolare una retta attorno ad una circonferenza. Un punto qualsiasi di tale retta si muoverà, appunto, secondo una traiettoria detta ad evolvente). Così costruiti, i denti "rotoleranno" fra loro e di conseguenza l'attrito sarà notevolmente ridotto.

Si definisce poi passo la distanza tra gli assi di due denti consecutivi misurata sulla circonferenza primitiva, valore che corrisponde allo spessore del dente e del vano adiacente.

Detto "z" il numero di denti, il passo varrà:

$$P = 3.14*Dp/z$$

E' chiaro che essendoci di mezzo un pigreco (3.141..) questo dovrebbe essere espresso con un numero irrazionale. Si preferisce per comodità usare come parametro caratteristico il cosiddetto Modulo dato da:

m = Dp/z

Il modulo non è altro che il passo diviso per pigreco. Una volta consciuto il modulo, la dentatura è completamente determinata. Infatti sia l'altezza che la larghezza del dente sono unicamente in funzione di quest'ultimo:

h. = 13/6*m

larg. = (da 8 a 10)*m

(= spessore consigliato)

questo vale ovviamente per dentature unificate. Rapporto di trasmissione (R). Quando abbiamo un motore che gira ad una certa velocità (pensiamo al motore dell'auto), può sorgere il problema che il numero di giri a disposizione non sia quello desiderato.

Per avere all'utilizzo (in questo caso l'albero di trasmissione) un numero di giri diverso, bisogna operare sulla scelta dei diametri, avvalendoci delle seguenti relazioni:

Vm/vc = Dc/dm = Zc/zm = R

Dove:

Vm = velocità ruota motrice (= che genera il movimento).

Dm = diametro primitivo ruota motrice.

Zm = numero di denti ruota motrice.

Vc = velocità ruota condotta (= trascinata dalla motrice).

Dc = diametro primitivo ruota condotta.

Zc = numero denti ruota condotta.

Si definisce quindi il rapporto di trasmissione R come il rapporto fra le velocità o i diametri o il numero di denti delle due ruote.

Il primo listato

Prendiamo in considerazione il caso più generale che può capitare in pratica:

- Conosciamo la velocità del motore e quella che vogliamo ottenere.
- Abbiamo delle dimensioni da rispettare quali interasse e ingombro (vedi figura 1). Si pensi alla scatola del cambio dove le dimensioni sono solitamente predeterminate e devono essere il più possibile contenute. Dato il RUN, il programma chiede di immettere l'interasse e il rapporto di trasmissione.

Quindi possiamo scegliere se far calcolare al programma stesso il valore del modulo, oppure se vogliamo immetterlo noi. Naturalmente questo valore deve essere proporzionale al diametro delle ruote, in quanto all'aumentare del modulo diminuisce il numero di denti, secondo la relazione:

Z = Dp/m

E' opportuno precisare che il numero di denti non può scendere al di sotto di certi valori (da 10 a 15) per evitare un funzionamento discontinuo e rumoroso. Nel caso si voglia calcolare il valore del modulo, il programma individuerà il minor numero di denti possibile. Da questo riceverà, con l'ausilio di una tabella, il valore unificato che più si avvicina a quello calcolato.

E' anche possibile immettere il dato direttamente. In questo caso verrà mostrata la tabella dei valori tra i quali operare la scelta.

Un esempio

Come valori iniziali battiamo (misure in mm.):

Interasse = 300 (R)

Numeratore del rapporto di trasmissione: 3 (R)

Denominatore: 5 (R)

Rispondiamo No alla richiesta di immissione del modulo, dopodichè verranno mostrate 5 diverse possibili soluzioni. La prima rappresenta il caso con il minor numero di denti possibile teoricamente. I primi due dati in alto sono rispettivamente l'interasse e il rapporto di trasmissione di partenza, che probabilmente si avvicineranno ma non coincideranno esattamente con i valori calcolati.

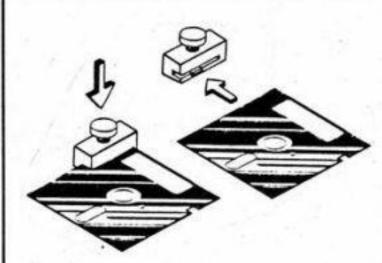
Questo si spiega col fatto che i moduli unificati sono solo una quarantina e l'intervallo esistente tra un valore e il successivo dà luogo a inevitabili

NEW SOFT S.R.L.

Accessori per Computer

Via Carbone, 8 - Tel. 0187/674097
19033 Castelnuovo Magra (SP)

Nastri per stampante

	Prezzo
Commodore MPS 801	**11.000**
Commodore MPS 802, Tally 80	**12.000**
Commodore MPS 803	**14.500**
Commodore 8024	5.100
Epson MX70,80,82,83,ERC-04, FX80, RX80, FX80, Commodore 4022, 8022, IBM P/C, Sharp CE332P, MZ 80P5A, PC3201	8.200
Commodore 3022, 3023, Epson TX80, Itoh 8300R, OKI 80, 82A, 83A, 92, 93, Sharp P3	3.000
Epson MX100	9.900
Commodore 8023P, MPP 1361 Sharp 80P4A, Centronics 150	8.950
Commodore 8026, 8027, 8032	6.950
Dischetti SF/DD x 10 (con box trasparente)	38.000
Dischetti DF/DD x 10 (con box trasparente)	43.000
Disco per pulizia delle testine. Questo può essere usato per drive con una o due facce. Il liquido basta per circa 15 applicazioni	12.200

Usate la seconda faccia del V/S Mini disco. Tagliate a metà il costo dei Dischetti! Foratore di Dischetti per usare anche l'altra faccia del disco. Per esempio Commodore 20/64, Apple 4, Atari, ecc. a sole ... 12.700

Tutti i prezzi sono IVA inclusa

Pagamento contrassegno. Per ordini superiori a L. 50.000 spese postali a nostro carico.

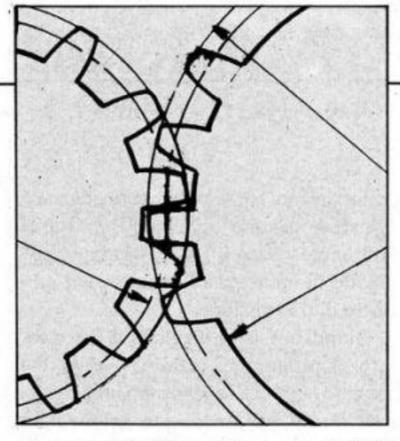
SPECIALI SCONTI A TUTTI I RIVENDITORI

arrotondamenti.

Ad esempio la n. 2 avrà come valori: 301 mm ed R = 1/1.69 contro i 300 mm ed R = 3/5 = 1/1.67 iniziali.

Come si può vedere l'approssimazione è abbastanza buona e molto probabilmente compatibile con i dati costruttivi. Fatta la scelta leggeremo nella pagina successiva tutti i dati che interessano per la completa definizione del rotismo.

Il secondo listato

Con il secondo programma è possibile calcolare la partenza massima che può essere trasmessa dall'ingranaggio in esame. Questa è funzione del modulo, del diametro primitivo e del numero di giri di funzionamento.

Gli ultimi due parametri influiscono sulla velocità periferica della ruota che è molto importante conoscere, per determinare il carico di sicurezza del materiale scelto.

Il programma infatti mostrerà un menù con i diversi materiali usati abitualmente, e chiederà in input il valore del carico di sicurezza statico (Kg/mm↑2). Questo verrà poi ricalcolato in funzione della velocità periferica, dato che questa aumenta le sollecitazioni indotte sui rotismi.

Verrà infine mostrata la potenza (in cavalli) che può essere trasmessa a varie velocità di rotazione. I criteri di calcolo possono variare anche notevolmente.

Comunque le formule usate sono quelle date normalmente nei manuali di meccanica.

Flavio Molinari

```
100 REM  ***********************
110 REM  *                     *
120 REM  *    COMMODORE 64     *
130 REM  *                     *
140 REM  *  RUOTE  DENTATE     *
150 REM  *                     *
160 REM  * BY FLAVIO MOLINARI  *
170 REM  *                     *
180 REM  ***********************
200 REM  *** TABELLA MODULI UNI *
    **
210 DATA  .5,.75,1,1.125,1.25,1.3
    75,1.5,1.75,2,2.25,2.5,2.75
220 DATA  3,3.25,3.5,3.75,4,4.5,5
    ,5.5,6,6.5,7,8,9,10
230 DATA  11,12,14,16,18,20,22,25
    ,28,32,36,40,45,50
240 DIM  MO(40):FOR  Q=1  TO  40:
    READ  M:MO(Q)=M:NEXT
260 REM  ***          INPUT       *
270 PRINT"[CLEAR]PROPORZIONAMENTO
     DELLE RUOTE DENTATE"
280 PRINT"CILINDRICHE SECONDO NOR
    ME UNI."
285 PRINT"TUTTE LE DIMENSIONI SON
    O ESPRESSE IN MILLIMETRI.":PR
    INT
290 INPUT "INTERASSE= MM";IT:IE=I
    T:PRINT
300 PRINT"RAPPORTO DI TRASMISSION
    E (A/B):"
302 INPUT "NUMERATORE (A)";A
304 INPUT "DENOMINATORE (B)";B
310 R=A/B:IF  R<1  THEN  R=B/A
315 X=R:GOSUB  860:R=X:RA=R
320 PRINT:PRINT"VUOI IMPOSTARE IL
     MODULO ? (S/N)"
330 GET  Q$:IF  Q$="S"  THEN  500
340 IF  Q$<>"N"  THEN  330
360 REM  MINIMO NUMERO DI DENTI
370 S=SIN(20*π/180)↑2:Z1=2/(SQR(R
    *R+(1+2*R)*S)-R)
380 REM  DIAMETRI E MODULO APPROS
    SIMATI
390 D1=2*IT/(1+R):D2=D1*R:M=D1/Z1
400 Q=1:REM  RICERCA MODULI UNIFI
    CATI
402 IF  M>50  THEN  Q=40:GOTO  42
    0
404 IF  M< 1  THEN  Q= 5:GOTO  42
    0
410 IF  M>MO(Q)  THEN  Q=Q+1:GOTO
     410
420 FOR  T=1  TO  5:M(T)=MO(Q-T+1
    ):NEXT
450 GOSUB  970:FOR  Q=1  TO  5:M=
    M(Q):GOSUB  800
460 PRINT"[RVS]SOLUZIONE"Q"[RVOFF
    ]";:GOSUB  880:NEXT:GOSUB  92
    0:END
500 PRINT"QUESTI SONO I MODULI UN
    IFICATI:":PRINT
510 FOR  Q=1  TO  40:PRINT MO(Q),
    :NEXT:PRINT:INPUT "SCEGLI";M
515 D1=2*IT/(1+R):D2=D1*R
520 GOSUB  970:GOSUB  800:GOSUB
```

```
    928:END
800 REM  *** DIMENSIONAMENTO ***
810 Z1=INT(D1/M+.5):Z2=INT(D2/M+.
    5):REM  NUMERO DENTI
820 P1=M*Z1:P2=M*Z2:T1=P1+2*M:T2=
    P2+2*M
830 B1=P1-14/6*M:X=B1:GOSUB  860:
    B1=X
840 B2=P2-14/6*M:X=B2:GOSUB  860:
    B2=X
850 IT=(P1+P2)/2:IG=T1+T2-2*M:REM
      INTERASSE E INGOMBRO
855 R=Z2/Z1:X=R:GOSUB  860:R=X:RE
    TURN
860 X=INT(X*100+.5)/100:RETURN:RE
    M  ARROTONDAMENTO
880 REM  ***    STAMPA     ***
890 PRINT  TAB(14)"MODULO="M; TAB
    (26)"R=1/"R:PRINT
900 PRINT"INTERASSE="IT; TAB(20)"
    INGOMBRO="IG
910 PRINT"------------------------
    ------------------":RETURN
920 REM  *** SCELTA ROTISMO ***
922 PRINT"QUALE SCEGLI ? (1-5)"
924 GET  Q$:IF  Q$<"1"  OR  Q$>"5
    "  THEN  924
926 M=M(VAL(Q$)):GOSUB  800:GOSUB
      970
928 PRINT:PRINT"RUOTA 1:":PRINT
929 PRINT"DIAM. PRIM. ="P1;" ** N
    . DENTI="Z1
930 PRINT"DIAM. DI TESTA="T1:PRIN
    T"DIAM. DI BASE="B1
932 PRINT:PRINT"RUOTA 2:":PRINT
933 PRINT"DIAM. PRIM. ="P2;" ** N
    . DENTI="Z2
934 PRINT"DIAM. DI TESTA="T2:PRIN
    T"DIAM. DI BASE="B2:PRINT
936 PRINT"MODULO="M:PRINT"PASSO="
    ;:PA=M*π:X=PA:GOSUB  860:PA=X
    :PRINT PA
938 PRINT"ALTEZZA DENTE=";INT(13/
    6*M*100+.5)/100
940 PRINT"INTERASSE="IT:PRINT"ING
    OMBRO="IG
942 PRINT"RAPP. DI TRASMISS.= 1/"
    R:RETURN
970 REM  *** VALORI INIZIALI ***
980 PRINT"[CLEAR]INTERASSE TEORIC
    O="IE
990 PRINT"RAPPORTO TEORICO R="A"/
    "B"=1/"RA
1000 PRINT"-----------------------
     ------------------":RETURN
```

RUOTE DENTATE DUE

```
2000 PRINT"[CLEAR]CALCOLO DELLA PO
     TENZA TRASMISSIBILE PER RUOTE
      DENTATE"
2005 PRINT:INPUT "MODULO = MM (0.5
     -50) ";M:INPUT "DIAMETRO PRIM
     ITIVO = MM";D
2010 PRINT:PRINT"[RVS]MATERIALI :"
     :PRINT
2020 PRINT"MATERIALE PLASTICO  KS=
     2 KG/MM↑2"
2030 PRINT"ALLUMINIO           KS=
     3 "
2040 PRINT"GHISA               KS=
     5"
2050 PRINT"BRONZO FOSFOROSO    KS=
     6"
2060 PRINT"ACCIAIO AL CARBONIO KS=
     10"
2070 PRINT"ACCIAIO AL NIKEL    KS=
     18"
2080 PRINT"ACCIAIO AL NIKEL-CROMO
            KS=25"
2090 PRINT"ACCIAIO AL NI-CR ALTA R
     ESIST. KS=50"
2100 KS$="CARICO DI SICUREZZA KS "
2105 PRINT:PRINT"SCEGLI IL "KS$;:I
     NPUT  KS
2110 N(1)=500:N(2)=2000:N(3)=5000:
     N(4)=8000:REM  NUMERO DI GIRI
2115 PRINT"[CLEAR]"KS$"="KS:PRINT"
     MODULO="M:PRINT"DIAM. PRIMIT.
     ="D:PRINT
2117 PRINT"POTENZA MAX TRASMISIBIL
     E A :":PRINT
2120 FOR  Q=1  TO  4:REM  ELABORAZ
     IONE
2130 V=π*D*N(Q)/60000:REM  VELOCIT
     A' PERIFERICA
2140 KD=KS*(3/(3+V)):REM  CARICO D
     I SICUR. DINAMICO
2150 P=M*M*KD*D*N(Q)/716200:P=INT(
     P*100+.5)/100:REM  POTENZA
2200 PRINT N(Q)"GIRI/MINUTO="P"CV"
     :NEXT:END
```

TI TELEFONO COL 64

di Gloriano Rossi

Un'agenda, un kit, una segretaria: TE combina mirabilmente il telefono ed il computer per darti tutto questo.

Agli albori dei personal ed home computers era d'uso porsi la domanda: quali saranno i limiti di queste macchinette?

Le risposte erano molteplici. Chi disquisiva sulle limitazioni dell'hardware, chi invece istaurava discorsi prolissi sulle caratteristiche dei linguaggi. Nessuno o pochi però avevano veramente individuato la vera risposta al problema.

La fantasia umana

Fa ormai parte della storia, e noi italiani abbiamo la nostra fetta di storia da rivendicare. Da quando nel 1977 vide la luce il primo personal computer veramente tale, il Commodore PET2001 (old ROM) incominciò l'invasione nelle case private. Di cosa? Di software, naturalmente. Fantasioso, bizzarro, improbabile, geniale: ogni genere di idea. Giochi, programmi didattici, utilità. O gestionali. Proprio in quest'ultima fascia incontriamo i tentativi più numerosi.

Una applicazione di un certo valore dipende da due fattori estremamente importanti: la capacità realizzativa e la professionalità.

Su molte riviste di settore, non ultima sulla consorella "Commodore", sono apparsi in passato articoli sui possibili utilizzi del telefono collegato al computer. Argomento peraltro piuttosto delicato. E' infatti tassativamente proibito manomettere in qualsiasi modo un apparecchio telefonico, tantomento la linea di comunicazione SIP. Ogni applicazione suggerita dalle riviste può essere studiata esclusivamente in via sperimentale e comunque non sulle linee ufficiali, pena sanzioni o revoca del contratto con l'azienda telefonica. A conferma di ciò ognuno di noi può documentarsi leggendo nelle prime pagine dell'elenco abbonati il regolamento ministeriale.

Ogni progetto deve superare esami severissimi, prima di ricevere l'omologazione da parte della Sip. Ecco quindi tornare il concetto di capacità e professionalità di una azienda.

La società Como Computer ha realizzato un sistema che è stato ritenuto valido dal Ministero delle Poste e Telecomunicazioni e quindi omologato nell'uso. Un successo che ribadisce le nostre capacità creative e che dimostra ulteriormente come le software house italiane nulla abbiano da invidiare ai colossi del software d'oltreoceano.

Quali sono a questo punto le caratteristiche che sono riuscite, per loro qualità, a superare il difficile ostacolo ministeria-

le? E' presto detto.

Il "telefono Elettronico" della Como Computer è un sistema misto: costituito da una parte hardware e da una parte software.

L'hardware

L'elettronica del sistema "TE" della Como Computer è estremamente semplice. Da una parte uno scatolino da connettere al C64 serve proprio da interfaccia fra il computer e la linea telefonica. Dall'altra una spina, chiamata maschio femmina, da mettere tra la presa SIP e la spina del telefono. La semplicità gia a questo punto è totale: nessuno sicuramente potrà mai lamentare difficoltà d'installazione.

La configurazione hardware necessaria è quindi sintetizzabile in:

- un telefono a spina (di cui si abbia regolarmente pagato la relativa bolletta);
- un Kit del "Telefono Elettronico" della Como Computer;
- un computer Commodore 64 (fra non molto sarà disponibile una versione di "TE" per l'Executive 64 ed in seguito una release per il Plus4);
- un floppy disc drive Commodore 1541;
- un monitor monocromatico o colore.

Il software

Il software del "TE" è la migliore sintesi di ciò che è comunemente conosciuto con il nome di Time Manager, un programma che gira normalmente sotto sistema operativo CP/M che ci permette di organizzare, nella maniera più opportuna, il nostro tempo.

Una vera agenda super gestita magistralmente dal C64.

Nel kit "TE" è presente un dischetto che contiene appunto il software che permette una facile gestione di ben 700 nominativi (per curiosità personale ho voluto controllare quanti fossero i numeri telefonici delle mie rubriche: poco più di 200!).

Ogni nominativo occupa su disco poco più di 370 caratteri così organizzati:

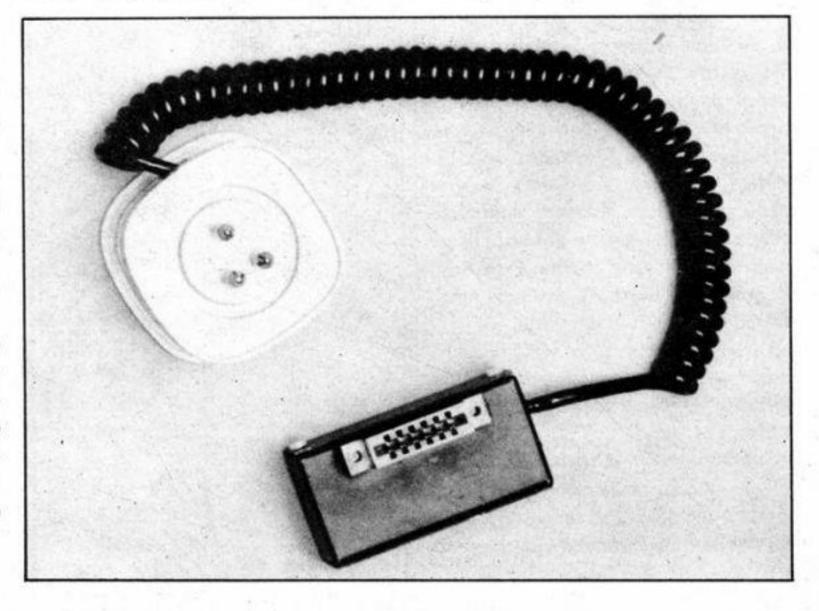
- Ragione sociale
- Data del prossimo appuntamento, espresso nella forma GGMMAA (giorno, mese e anno)
- Fascia oraria di presenza
- Numero telefonico. (Il numero telefonico è comprensivo del prefisso, del numero vero e proprio ed infine dei tempi di pausa programmabili)
- Indirizzo
- Codice di avviamento postale
- Località
- Cumulo del numero delle telefonate effettuate con quel corrispondente
- Cumulo dei tempi e/o scatti delle telefonate con quel corrispondente
- Data dell'ultima telefonate effettuata con quel corrispondente
- Cumulo dei tempi e/o scatti delle telefonte effettuate con quel corrispondente
- Note varie inseribili in un totale di sei righe di schermo per un totale di 240 caratteri.

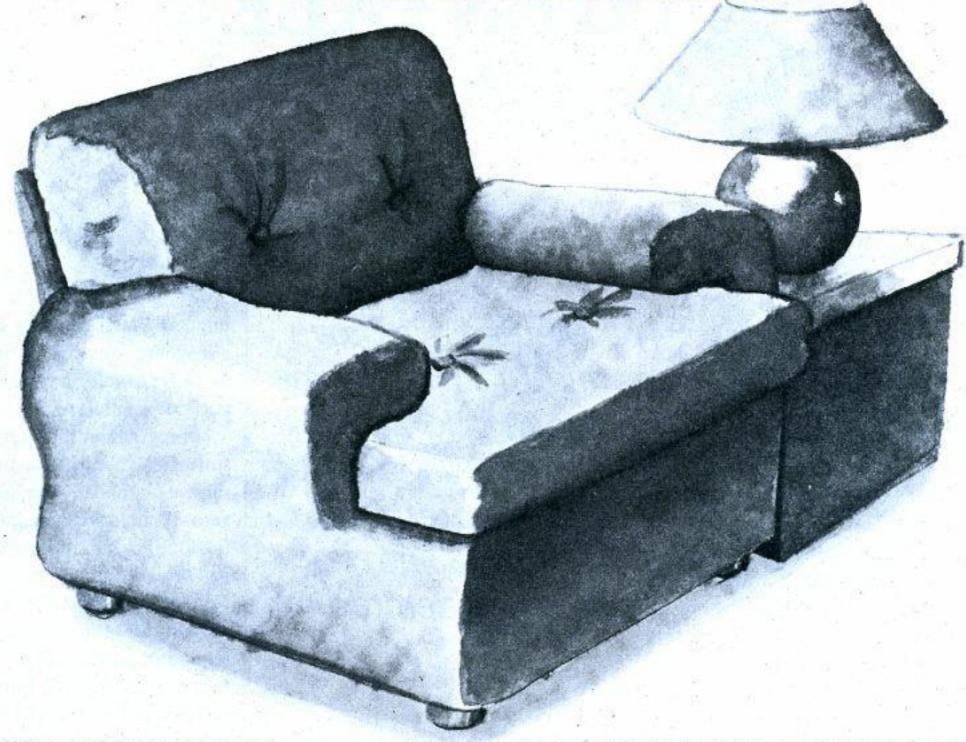
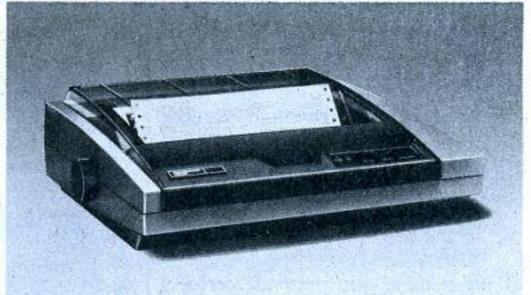
Nell'archivio principale ogni nominativo può essere creato, modificato o cancellato facilmente. Si potrà quindi aggiornare qualsiasi caratteristica: dal numero telefonico alla ragione sociale, dall'indirizzo all'azzeramento dei contatori. Sul dischetto lavoro sono presenti altri archivi, più o meno trasparenti, che permettono una più ampia possibilità di utilizzazione del sistema "TE". Ad esempio l'archivio "lista di attesa" o quello di "memoria".

L'operatività di TE

Quando il programma "TE" è presente in memoria, il sistema si informa: chiede data e ora. Quindi rimane in attesa operativa. Da qui in poi si potrà interagire con l'archivio principale, ricercare un nominativo, comandare al "TE" di comporre un numero di telefono, porlo in attesa, memorizzarlo per un prossimo appuntamento, etc. In ogni momento l'utente può informarsi su quali appuntamenti, telefonici o non, ha in quel determinato giorno ed in quale fascia oraria;un nominativo può essere chiamato in maniera automatica o semiautomatica. .

In sintesi l'utente ha per le mani la più efficente segretaria telefonica esistente.

A questo punto mi sorge un sospetto: si potrà comandare al "TE" (avete notato che leggendo al contrario...) di farmi da svegla? Potrei ad esempio mantenere in funzione il sistema in ufficio, impostare il mio numero di casa con relativo orario per la chiamata. E TE allo scoccare dell'ora definita mi telefonerebbe e mi farebbe proprio da sveglia telefonica. Spendendo solamente uno scatto e forse nemmeno quello!

Le proposte del n. 1

PER STAMPARE A CASA TUA

La MT/80+ e la PC dispongono di interfacciamento parallelo e seriale che permette di connettere questi prodotti a qualsiasi Micro o PC. Per la stampa a basso costo, le stampanti della famiglia MT/80 sono perfette, rispondendo ad ogni tua necessità di stampa.
La velocità di stampa é a 100 o 130 cps. con una matrice estremamente chiara e pertanto ideale per stampare i tuoi listati, le tue lettere ecc.
Le stampanti della famiglia MT/80 oltre che trascinare la carta con trattori di spinta consentono anche, per mezzo della frizione, di inserire il foglio singolo.
Le MT/80 sono belle a vedersi, facili da usare, non richiedono manutenzione preventiva, silenziose ed in più la versione 80 PC é completamente compatibile con il tuo PC IBM.

20094 Corsico (MI) - Via Cadamosto, 3
Tel. (02) 4502850/855/860/865/870 - Telex 311371 Tally I
00137 Roma - Via I. Del Lungo, 42 - Tel. (06) 8278458
10099 San Mauro (TO) - Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO) - Via Einstein, 5 - Tel. (051) 832508

adverteam

STRESS MENTALE

Mettete alla prova riflessi e nervi in una prova di forza con il computer. Ma senza esagerare: lui, alla fine, vince sempre.

Nello studio delle malattie da ipertensione o durante un normale esame di medicina sportiva, i pazienti vengono sottoposti ad una prova di sforzo durante la quale il ritmo cardiaco e la pressione vengono continuamente monitorati.

La prova di sforzo può essere esclusivamente fisica, (pedalare su un'apposita ciclette), oppure anche mentale. Uno dei modi per sottoporre un paziente ad uno stress mentale è quello di costringerlo a decrementare una grossa cifra sottraendo in continuazione un numero qualunque. Il tutto naturalmente deve essere effettuato in un tempo limite altrimenti addio stress!

Durante l'esame un medico controlla su una tabella i risultati delle sottrazioni effettuate dal paziente (la tabella è necessaria per non mandare in stress anche il medico), mentre un altro paio di medici controllano la pressione e il tracciato dell'elettro-cardio-gramma del paziente durante la prova stessa.

Il programma non pretende certo di effettuare un esame vero e proprio ma ad esso si ispira per permettere ad ognuno di noi di confrontare la propria velocità di calcolo con quella degli amici.

Un consiglio è però doveroso: non insistete molto nel tentare di migliorare il vostro punteggio, altrimenti lo stress comincerà a farsi sentire, e allora inizierete a sbagliare con maggior frequenza facendo calare di volta in volta il punteggio che volevate migliorare.

Il programma infatti assegna ad ogni sottrazione ben fatta un punteggio inversamente proporzionale al tempo impiegato. Il punteggio massimo per ogni sottrazione è 10, ma dopo un secondo è già passato a 9 poi 8 e così via. Se si arriva sotto lo zero sarà il computer stesso a darvi il risultato ma decrementerà il vostro punteggio totale di 10 unità. Più velocemente svolgete le sottrazioni, più velocemente il punteggio sale; se invece superate un tempo limite di 10 secondi per ciascuna sottrazione il punteggio calerà paurosamente.

Naturalmente per il conteggio dei dieci secondi è stato utilizzato l'orologio interno, mentre per la visualizzazione delle cifre sono stati utilizzati maxi-caratteri costruiti attraverso l'uso dei caratteri semigrafici.

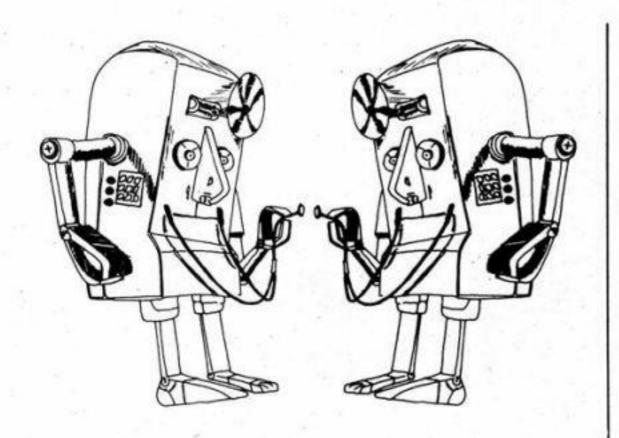
Per una più facile battitura del programma, a fianco di ogni istrusione di stampa, sono stati specificati i tasti da premere per ottenere i caratteri semigrafici (C= tasto Commodore in basso a sinistra sulla tastiera). Non stressatevi troppo!

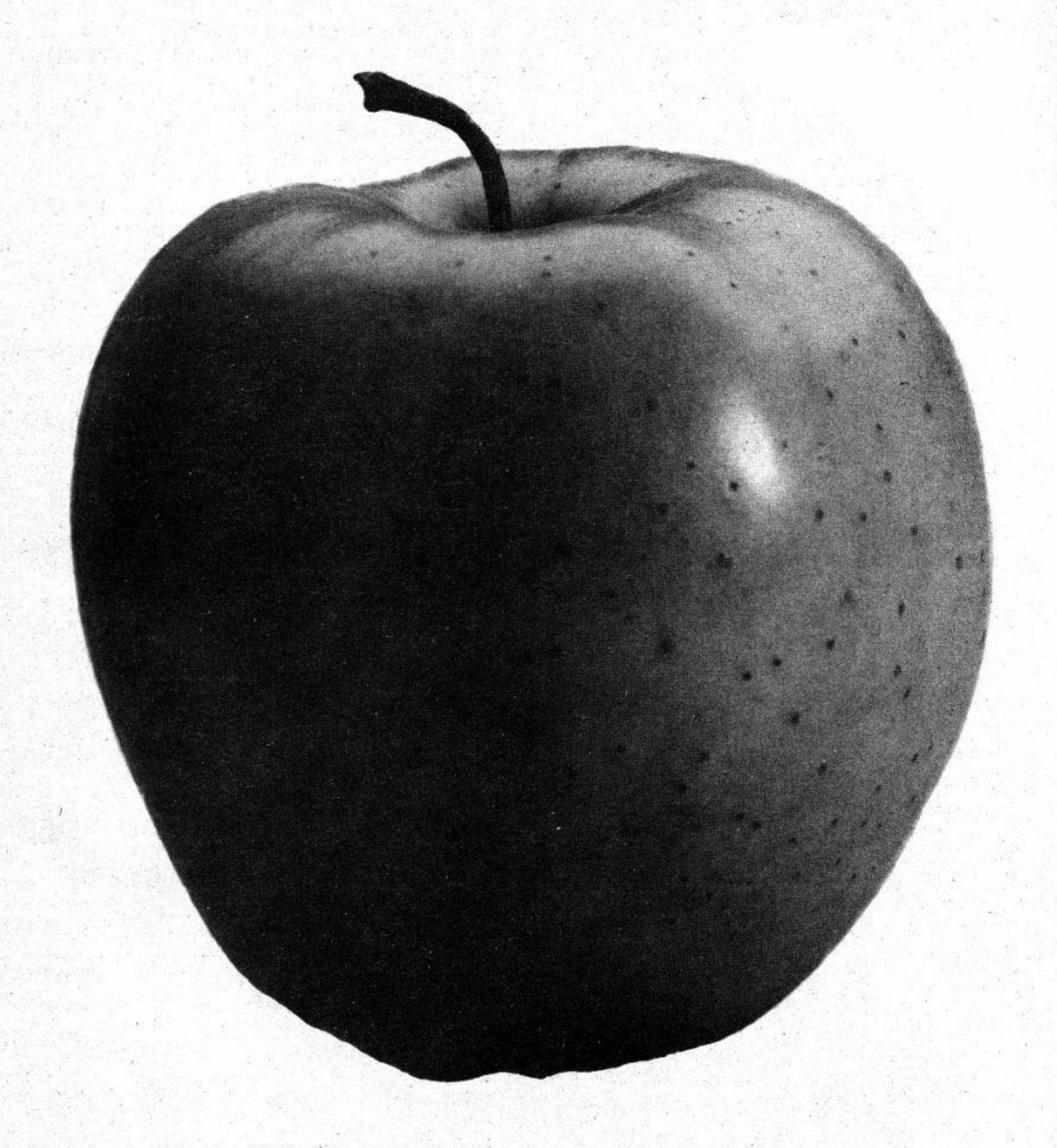
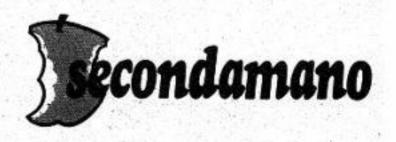
Luca Galuzzi

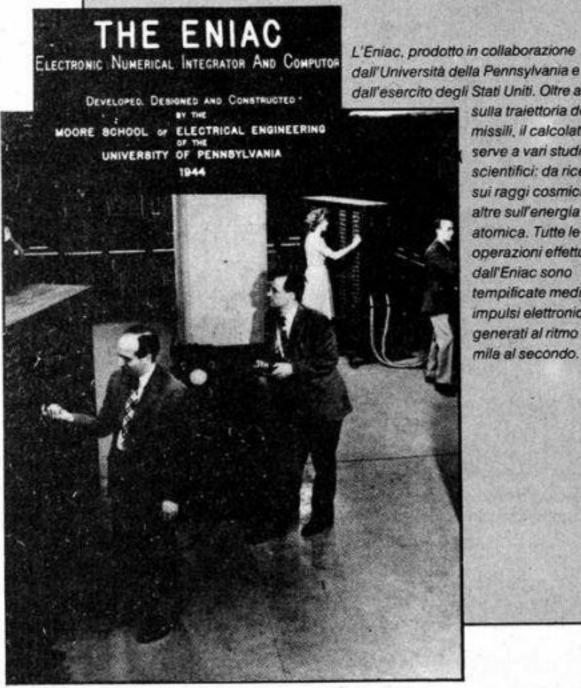
Le proposte del n. 1

PER STAMPARE CON POCHE LIRE

Le stampanti **MT/85, a 80 colonne, e MT/86, a 136 colonne,** rappresentano una nuova frontiera nel settore delle stampanti a basso costo.
Basso costo, ma non bassa qualità e basse prestazioni, infatti ecco le credenziali di questi due nuovi prodotti.
Velocità a 180 cps. bidirezionale ottimizzata, NLQ a 45 cps., grafiche, possibilità di 8 fonti alternative di caratteri e naturalmente la completa compatibilità con il PC IBM.
Il prezzo: il più competitivo del mercato in questa fascia di prestazioni.
Naturalmente anche le MT/85/86 oltre ai trattori hanno anche trascinamento a frizione e consentono pertanto il trattamento del foglio singolo.

20094 Corsico (MI) - Via Cadamosto, 3
Tel. (02) 4502850/855/860/865/870 - Telex 311371 Tally I
00137 Roma - Via I. Del Lungo, 42 - Tel. (06) 8278458
10099 San Mauro (TO) - Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO) - Via Einstein, 5 - Tel. (051) 832508

```
 10 REM  C-16 VIC 20
 20 REM  COMMODORE 64
 21 REM  PLUS 4
 22 :
 25 REM  - STRESS MENTALE -
 30 :
 35 REM  DI LUCA GALUZZI
 90 :
 95 X1=0:X2=0:PRINT CHR$(147)"1- V
    IC 20 & C-64"
 96 PRINT"2- C-16 & PLUS 4"
 97 GET  GG$:IF GG$=""  THEN  97
 98 GG=198:IF GG$="2" THEN GG=239
 99 FOR I=1 TO 4:G(I)=630+I:NEXT
100 IF  GG=239  THEN  FOR I=1 TO 4
    :G(I)=1318+I:NEXT
105 POKE  GG,0
106 XX$="":FOR I=1 TO 10:XX$=XX$+C
    HR$(17):NEXT:XX$=CHR$(19)+XX$
110 PRINT CHR$(147) "IL SEGUENTE
    TEST  PERMETTE DI MISURARE"
120 PRINT"QUANTITATIVAMENTE    LA
    VOSTRA  SOPPOR-"
130 PRINT"TAZIONE DI UNO STRESS ME
    NTALE PROTRATTO"
140 PRINT"PER  QUALCHE  MINUTO. PE
    RMETTE NEL CON-"
150 PRINT"TEMPO  DI AUMENTARE  LA
    VOSTRA AGILITA'"
160 PRINT"MENTALE DI CALCOLO."
170 PRINT"IL  COMPUTER  VI PROPORR
    A' UN NUMERO DI"
180 PRINT"QUATTRO  CIFRE  DAL  QUA
    LE DOVRETE SOT-"
190 PRINT"TRARRE  IN CONTINUAZIONE
    UN NUMERO DATO"
200 PRINT"DI  DUE CIFRE, PER UN NU
    MERO DI VOLTE"
210 PRINT"SCELTO ALL'INIZIO."
220 PRINT"                    TUTT
    O QUESTO DOVRA'"
230 PRINT"ESSERE  FATTO NEL MINOR
    TEMPO POSSIBILE"
240 PRINT"POICHE' IL  PUNTEGGIO  C
    HE VIENE ATTRI-"
250 PRINT"BUITO  AD OGNI SOTTRAZIO
    NE E' CONTINUA-"
260 PRINT"MENTE  DECREMENTATO, QUI
    NDI PIU' VELOCE"
270 PRINT"SARA' LA  RISPOSTA MAGGI
    OR PUNTEGGIO VI"
280 PRINT"SARA' ATTRIBUITO."
285 INPUT "QUANTE VOLTE (MIN.99)";
    X1
290 PRINT"[2 DOWN]    PREMI UN TAS
    TO PER CONTINUARE"
300 GET A$:IF A$="" THEN  300
305 PRINT CHR$(147)
310 N=INT(RND(1)*7000)+2000
320 M=INT(RND(1)*89)+10
330 IF M/10=INT(M/10) THEN 320
340 GOTO 800
400 REM  *************************
    ***
410 REM  ***   STAMPA  NUMERI
    ***
420 REM  *************************
    ***
490 IF X1<=X2 THEN 10000
500 PRINT CHR$(19)"
                        ":REM  36
     SPAZI
510 PRINT CHR$(19)"PUNTEGGIO ";P;"
     --- TOTALE ";T;
520 A=10:G=INT(N/1000):H=G
530 GOSUB  9000
540 A=15:G=INT(N/100)-(G*10):L=G
550 GOSUB  9000
560 A=20:G=INT(N/10)-(G*10)-(H*100
    )
570 GOSUB  9000
580 A=25:G=N-(G*10)-(L*100)-(H*100
    0)
590 GOSUB  9000
600 PRINT XX$
610 A=15:G=INT(M/10)
```

Caro piano media, prendi le vitamine.

Secondamano, il media polivitaminico più indicato in Lombardia. (Vitamina B) 70mila copie bisettimanali - (Vitamina C) 700mila lettori, con aumento costante del 30% annuo - (Vitamine A, E, D) Network di 29 testate a inserzioni gratuite come copertura nazionale. Posologia e indicazioni: Usare spesso. L'azione tonificante delle 70mila copie e dei 700mila lettori è stata sperimentata con successo anche sui piani media più abulici. Controindicazioni: Nessuna.

```
620 GOSUB  9010
625 PRINT XX$
630 A=20:G=M-(G*10)
640 GOSUB  9010
650 PRINT XX$ CHR$(17) CHR$(17)
660 PRINT  TAB(11) "___":REM  C +
    P
670 RETURN
800 REM  *************************
    *
810 REM  ****    M  A  I  N    ***
    *
820 REM  *************************
    *
830 P=10:POKE  GG,0
840 TI$="000000"
842 GOSUB 490
845 P=P-INT(TI/100):IF TI>=100 THE
    N 840
850 IF P=0 THEN 950
870 IF PEEK(GG)<4 THEN 842
885 J=PEEK(G(4))-48+(PEEK(G(3))-48
    )*10
890 J=J+(PEEK(G(2))-48)*100+(PEEK(
    G(1))-48)*1000
```

Le immagini di questo fascicolo

L'Eniac, prodotto in collaborazione dall'Università della Pennsylvania e dall'esercito degli Stati Uniti. Oltre a calcoli sulla traiettoria dei missili, il calcolatore serve a vari studi scientifici: da ricerche sui raggi cosmici ad altre sull'energia atomica. Tutte le operazioni effettuate dall'Eniac sono tempificate mediante impulsi elettronici generati al ritmo di 100 mila al secondo.

```
905 POKE  GG,0:X2=X2+1
910 IF J=N-M THEN N=N-M:T=T+P:GOTO
    830
930 GOSUB  30030
940 GOTO 830
950 GOSUB 20030
960 GOTO 830
1000 PRINT  TAB(A)"/‾‾\":REM  SHIFT
     N ,C= Y ,C= Y ,SHIFT M
1010 PRINT  TAB(A)"|  |":REM  C= G
     ,C= M
1020 PRINT  TAB(A)"|  |":REM  C= G
     ,C= M
1030 PRINT  TAB(A)"\__/":REM  SHIFT
     M ,C= P ,C= P ,SHIFT N
1040 RETURN
1100 PRINT  TAB(A)" /| ":REM  SHIFT
     N ,C= G
1110 PRINT  TAB(A)"  | ":REM  C= G
1120 PRINT  TAB(A)"  | ":REM  C= G
1130 PRINT  TAB(A)" _|_":REM  C= P
     ,SHIFT L
1140 RETURN
1200 PRINT  TAB(A)"/‾‾\":REM  SHIFT
     N ,C= Y ,C= Y ,SHIFT M
1210 PRINT  TAB(A)" __/":REM  C= P
     ,C= P ,SHIFT N
1220 PRINT  TAB(A)"/   ":REM  SHIFT
     N
1230 PRINT  TAB(A)"L___":REM  SHIFT
     L ,C= P ,C= P ,C= P
1240 RETURN
1300 PRINT  TAB(A)"/‾‾\":REM  SHIFT
     N ,C= Y ,C= Y ,SHIFT M
1310 PRINT  TAB(A)" __/":REM  C= P
     ,C= P ,SHIFT N
1320 PRINT  TAB(A)"   \":REM  SHIFT
     M
1330 PRINT  TAB(A)"\__/":REM  SHIFT
     M ,C= P ,C= P ,SHIFT N
1340 RETURN
1400 PRINT  TAB(A)"  /|":REM  SHIFT
     N ,C= H
1410 PRINT  TAB(A)" / |":REM  SHIFT
     N ,C= H
1420 PRINT  TAB(A)"/  |":REM  SHIFT
     N ,C= H
1430 PRINT  TAB(A)"‾‾‾T":REM  C= Y,
     C= Y,C= Y,SHIFT O
1440 RETURN
1500 PRINT  TAB(A)"Г‾‾‾":REM  SHIFT
```

DIGITARE STANCA

I programmi più interessanti spesso sono molto lunghi, un listato pubblicato è faticoso da leggere...

Commodore Computer Club vi offre un'alternativa: le cassette con tutti i programmi pubblicati sulla rivista.

Ogni nastro contiene il software di un numero di Commodore Computer Club a un prezzo incredibilmente basso: solo 5.800 lire (+ 1.000 lire per spese di spedizione).

Riceverete le cassette direttamente a casa vostra, utilizzando il coupon qui a fianco.

Desidero ricevere le cassette con il software pubblicato sui seguenti numeri di Commodore Computer Club:

....

importo L. ....
spese di spedizione L. 1.000

Totale L. ....

☐ ho versato l'importo sul c/c postale n. 31532203 (allego fotocopia della ricevuta di versamento)
☐ accludo assegno non trasf. n. .... (banca ....) intestato a C.C.C, viale Famagosta, 75- 20142 Milano

nome ....

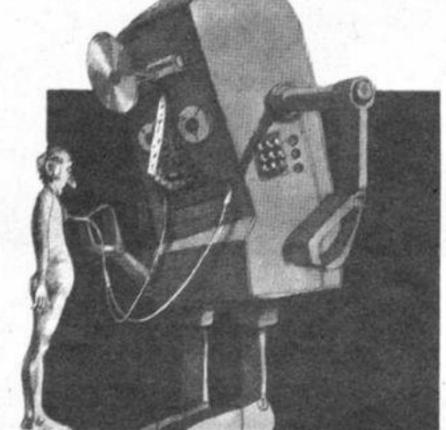
cognome ....

via ....

CAP/città ....

Ritagliare e spedire in busta chiusa a: Systems Editoriale v.le Famagosta 75, 20142 Milano.

Abbonatevi a Commodore Computer Club

```
     0 ,C= Y,C= Y,C= Y
1510 PRINT  TAB(A)"L__ ":REM  SHIFT
      L ,C= P,C= P
1520 PRINT  TAB(A)"   \":REM  SHIFT
      M
1530 PRINT  TAB(A)"\__/":REM  SHIFT
      M,C= P,C= P,SHIFT N
1540 RETURN
1600 PRINT  TAB(A)"/‾‾\":REM  SHIFT
      N,C= Y,C= Y,SHIFT M
1610 PRINT  TAB(A)"L__ ":REM  SHIFT
      L,C= P,C= P
1620 PRINT  TAB(A)"|  \":REM  C= H,
     SHIFT M
1630 PRINT  TAB(A)"\__/":REM  SHIFT
      M,C= P,C= P,SHIFT N
1640 RETURN
1700 PRINT  TAB(A)"‾‾‾]":REM  C= Y,
     C= Y,C= Y,SHIFT P
1710 PRINT  TAB(A)"   /":REM  SHIFT
      N
1720 PRINT  TAB(A)"  / ":REM  SHIFT
      N
1730 PRINT  TAB(A)" /  ":REM  SHIFT
      N
1740 RETURN
1800 PRINT  TAB(A)"/‾‾\":REM  SHIFT
      N,C= Y,C= Y,SHIFT M
1810 PRINT  TAB(A)"\__/":REM  SHIFT
      M,C= P,C= P,SHIFT N
1820 PRINT  TAB(A)"/  \":REM  SHIFT
      N,SHIFT M
1830 PRINT  TAB(A)"\__/":REM  SHIFT
      M,C= P,C= P,SHIFT N
1840 RETURN
1900 PRINT  TAB(A)"/‾‾\":REM  SHIFT
      N,C= Y,C= Y,SHIFT M
1910 PRINT  TAB(A)"\__]":REM  SHIFT
      M,C= P,C= P,SHIFT @
1920 PRINT  TAB(A)"   |":REM  C= N
1930 PRINT  TAB(A)"\__/":REM  SHIFT
      M,C= P,C= P,SHIFT N
1940 RETURN
8900 :
8310 REM  SCELTA CIFRA
8920 :
9000 PRINT CHR$(19) CHR$(17) CHR$(1
     7) CHR$(17) CHR$(17)
9010 ON G+1GOSUB 1000,1100,1200,130
     0,1400,1500,1600,1700,1800,190
     0
9020 RETURN
10000 :
10010 REM  FINE GIOCO
10020 :
10030 PRINT"[CLEAR][3 DOWN]HAI TOTAL
      IZZATO ";T;"PUNTI"
10035 PRINT"[2 DOWN]"
10040 IF T<0 THEN PRINT"ALLENATI ! S
      EI MOLTO SCARSO"
10050 PRINT"[4 DOWN]VUOI RICOMINCIAR
      E ?"
10055 POKE GG,0:T=0
10057 GET A$:IF A$="" THEN 10057
10060 IF A$="S" OR A$="Y" THEN 10
10070 PRINT"[CLEAR][2 DOWN]CIAO..."
10080 END
20000 :
20010 REM  TEMPO SCADUTO
20020 :
20030 IF  GG=198  THEN  POKE 53280,1
      :GOTO 20035
20035 FOR K=1 TO 500 :NEXT
20040 N=N-M
20050 T=T-10
20060 IF  GG=198  THEN  POKE 53280,0
      :RETURN
30000 :
30010 REM  ERRORE
30020 :
30030 IF  GG=198  THEN  POKE 53281,1
30035 FOR K=1 TO  500:NEXT
30040 N=N-M
30050 T=T-10
30060 IF  GG=198  THEN  POKE 53281,0
30070 RETURN
```

ABBATTI IL MURO

Mattone dopo mattone, colpo su colpo. Il gioco più semplice per il tuo nuovo C 16

Bisogna abbattere il muro, mattone dopo mattone, con una pallina. I rimbalzi della pallina, sono comandati da una specie di racchetta che può muoversi solo verso destra o verso sinistra. Questi movimenti, si possono ottenere premendo i tasti 'A' (destra) ed 'L' (sinistra).

Se la pallina colpisce la parte centrale della racchetta, rimbalza verticalmente, mentre se colpisce la parte destra o sinistra, rimbalza con un'inclinazione di 45 gradi (rispettivamente, verso destra o verso sinistra). I tentativi a disposizione sono 4 (quattro palline) ed ogni pallina è persa se scende al di sotto della racchetta.

Per iniziare il gioco, o per riprenderlo dopo la perdita di una pallina, occorre premere il tasto '£'. Alla fine del partita, premendo 'Y' si ricomincia, mentre premendo 'N' si ferma il programma.

Il programma, fa largo uso di istruzioni PEEK e POKE, tutte riferite alla memoria di schermo, che va da 3072 a 4071 (sullo schermo possono 'apparire', nello stesso istante, 1000 caratteri).

La variabile A1$ che compare in linea 210, si ottiene in questo modo: (tasto Commodore = C)

A1$="<C + 4> <CTRL + 9> <spazio> <CTRL + 0> <CTRL + +> <C + 3> <CTRL + 9> <spazio> <CTRL + 0> <CTRL + 1>"

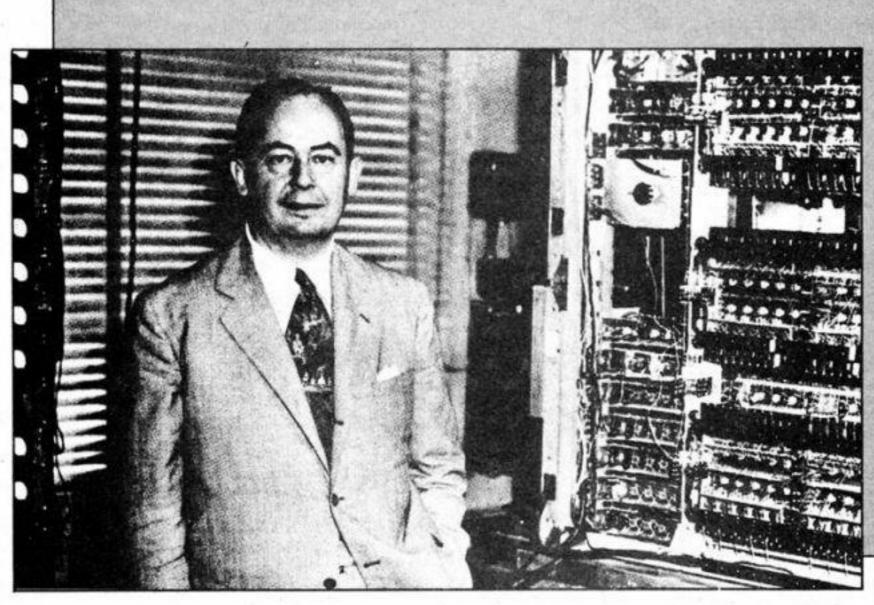
Per la variabile A2$, i simboli usati sono gli stessi.

Se i tasti di movimento 'A' e 'L' non vi sembrano 'comodi', potete cambiarli semplicemente modificando i caratteri tra 'virgolette' ("A" — "L") di linea 400 e 410.

Giancarlo Castagna

```
  1 REM  ************************
  2 REM  * COMMODORE C16         *
  3 REM  *                       *
  4 REM  *  ABBATTERE IL MURO!   *
  5 REM  *                       *
  6 REM  *   GIANCARLO CASTAGNA  *
  7 REM  ************************
  8 :
  9 :
100 PRINT CHR$(147): S1=3922: S2=39
    23: S3=3924: S4=3603
110 II=40: SP=37: FOR  I=3072  TO
    3151: POKE I,160: NEXT
120 B1$="[RVS]GAME OVER[RVOFF]":B2$
    ="[RVS]CAMPIONE[RVOFF]": PRINT
    CHR$(19)
130 PRINT SPC(4) "[RVS]ABBATTERE IL
     MURO![RVOFF]" SPC(6) "[RVS]@19
    85 CAST[RVOFF]"
140 PRINT  CHR$(17) CHR$(17) SPC(30
    ) "[RVS]PUNTI[RVOFF]"
150 PRINT CHR$(17) CHR$(17) CHR$(17
    ) SPC(29) "[RVS]PALLINE[RVOFF]"
160 PRINT CHR$(17) SPC(29) "● ● ● ●
    "
170 PRINT CHR$(17) CHR$(17) CHR$(17
    ) SPC(28) "[RVS]START '£'[RVOFF
    ]"
180 FOR  I=3152  TO  3191: POKE I,2
    32: POKE I+880,160: NEXT
190 FOR  I=3192  TO  3992  STEP  40
    : POKE I,220: POKE I+39,160
200 POKE I+25,220: NEXT: POKE 3191,
    160
210 A1$="[GRIGIO1][RVS] [RVOFF][NER
    O][ROSA][RVS] [RVOFF][NERO]": A
    2$="[ROSA][RVS] [RVOFF][NERO][G
    RIGIO1][RVS] [RVOFF][NERO]"
220 PRINT CHR$(19) CHR$(17) CHR$(17
    )
230 PRINT CHR$(29);: FOR  I=1  TO
    12: PRINT A1$;: NEXT: PRINT
240 PRINT CHR$(29);: FOR  I=1  TO
    12: PRINT A2$;: NEXT: PRINT
250 C=C+1: IF  C<4  THEN  230
260 FOR  I=3993  TO  4016: POKE I,1
    02: NEXT
270 PRINT CHR$(19): FOR  I=1  TO  9
    : PRINT: NEXT
280 SP=SP-2: PRINT SPC(SP) "  "
290 POKE S1,79: POKE S2,119: POKE S
    3,80: POKE S4,81
300 GET  Q$: IF  Q$<>"£"  THEN  300
310 GET  M$
320 S4=S4+II: POKE S4,81: POKE S4-I
    I,32
330 IF  PEEK(S4+40)=102  THEN  630
340 IF  PEEK(S4+II)=119  THEN  II=-
    40: GOSUB  620
350 IF  PEEK(S4+II)=80  THEN  II=-3
    9: GOSUB  620
360 IF  PEEK(S4+II)=79  THEN  II=-4
    1: GOSUB  620
370 IF  PEEK(S4-40)=160  THEN  GOSU
    B  500
380 IF  PEEK(S4-40)=232  THEN  GOSU
    B  590
390 IF  PEEK(S4+1)=220  OR  PEEK(S4
    -1)=220  THEN  GOSUB  550
400 IF  M$="A"  THEN  420
410 IF  M$="L"  THEN  460:  ELSE  3
    10
420 S1=S1-1: S2=S2-1: S3=S3-1
430 IF  S1<3913  THEN  S1=3913: S2=
    3914: S3=3915
440 POKE S1,79: POKE S2,119: POKE S
    3,80: POKE S3+1,32
450 GOTO  310
460 S1=S1+1: S2=S2+1: S3=S3+1
470 IF  S3>3936  THEN  S3=3936: S2=
    3935: S1=3934
480 POKE S1,79: POKE S2,119: POKE S
    3,80: POKE S1-1,32
```

```
490 GOTO  310
500 POKE S4-40,32
510 IF  II=-40  THEN  II=40
520 IF  II=-39  THEN  II=41
530 IF  II=-41  THEN  II=39
540 RETURN
550 IF  II=41  THEN  II=39: RETURN
560 IF  II=39  THEN  II=41: RETURN
570 IF  II=-39  THEN  II=-41: RETUR
    N
580 IF  II=-41  THEN  II=-39: RETUR
    N
590 IF  II=-40  THEN  II=40: RETURN
600 IF  II=-41  THEN  II=39: RETURN
610 IF  II=-39  THEN  II=41: RETURN
620  VOL8: SOUND1,500,10: RETURN
630  VOL8: SOUND3,500,10: H=0: PU=0
    : GOSUB  720
640 POKE S1,32: POKE S2,32: POKE S3
    ,32: POKE S4,32
650 S1=3922: S2=3923: S3=3924: S4=3
    603: II=40
660 B3$=B1$: B=B+1: IF  B<4  THEN
    270
670 FOR  I=1  TO  13: PRINT: NEXT:
    PRINT SPC(28) B3$
680 PRINT CHR$(17) SPC(31) "Y/N"
690 GET  R$: IF  R$="N"  THEN  END
700 IF  R$="Y"  THEN  710:  ELSE  6
    90
```

```
710 C=0: B=0: PU=0: GOTO  100
720 FOR  I=3193+H  TO  3216+H: IF
    PEEK(I)=32  THEN  PU=PU+1
730 NEXT: H=H+40: IF  H<320  THEN
    720
740 PRINT CHR$(19): FOR  I=1  TO  5
    : PRINT: NEXT
750 PRINT SPC(30) PU
760 IF  PU=192  THEN  B3$=B2$: GOTO
     670
770 RETURN
```

Le immagini di questo fascicolo

John von Neumann, scienziato di origine ungherese, e la sua macchina, l'EDVAC (vuol dire: computer elettronico a variabile automatica discreta) elaborata nel 1945 nell'Istituto di Studi Avanzati dell'Università di Princeton negli Stati Uniti. Il principale concetto introdotto da von Neumann è quello di "programma memorizzato". Il calcolatore diventa elaboratore: capace di eseguire istruzioni intermedie. E, soprattutto, di trattare qualsiasi informazione, non solo cifre.

GIOCHI
COMMODORE 64

IL RANOCCHIO AFFAMATO

La rana ha fame, e gli insetti sono vicini. Ma se sei bravo puoi farne un gioco ancora più bello.

Una rana affamata è a caccia di insetti, alcuni sprite multicolori che continuano a passare casualmente, in orizzontale sullo schermo. Il ranocchio è invece fatto di due o tre sprite, a seconda che si trovi a terra o in volo. Premendo un tasto qualsiasi, lui fa un balzo. E se l'insetto viene a contatto con la testa, ecco che sparisce nelle fauci dell'anfibio affamato.

E aumenta il punteggio.

Ad ogni balzo però il ranocchio fa un balzo in avanti, rendendo così più difficile la cattura dell'insetto.

A questo punto, il problema diventa unicamente di strategia, a voi il compito di risolverlo.

1053 PUNTI

Rispetto ad altri giochi simili, questo ha la particolarità di essere tanto semplice e chiaro da potere essere rivoluzionato e abbellito dai vostri interventi.

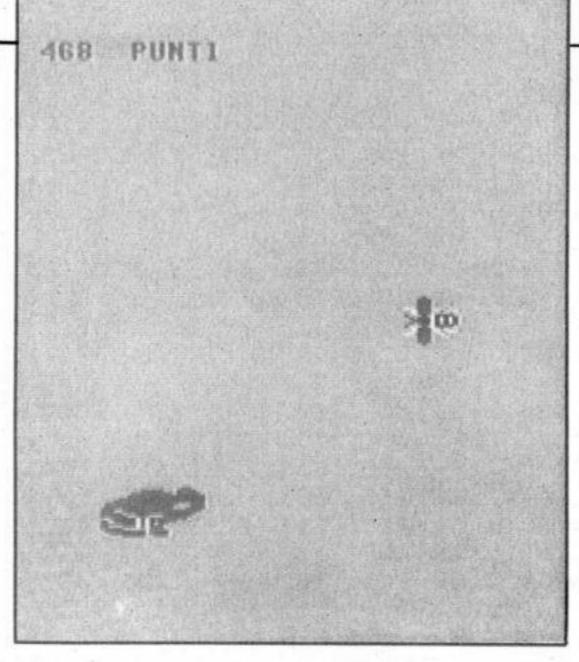
In particolare, potrete sbizzarrirvi nel rifare gli sprite multicolori e la loro animazione.

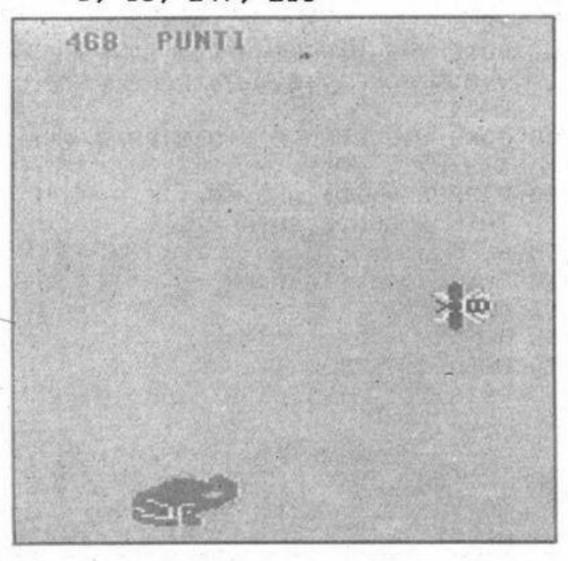
Abbandonando l'idea del ranocchio, a saltare potrebbe ad esempio essere un piccolo pac-man, una molla ingorda, un piccolo canguro.

La nostra proposta si trasforma così in una specie di canovaccio sul quale lavorare liberamente. Passo a passo, salto per salto.

Luca Galluzzi

Le immagini di questo fascicolo

E' una macchina gigantesca. 800 chilometri di fili elettrici collegano 78 calcolatrici. Contiene 3300 relè per muovere le parti meccaniche. E' il Mark 1, 5 tonnellate di peso, messo a punto nel 1944 all'università di Harvard. E' il primo calcolatore aritmetico di tipo universale. I dati numerici possono essere introdotti mediante nastro perforato o interruttori azionati manualmente (nel riquadro).

```
100 PRINT CHR$(147)
110 POKE  53281,6: POKE  53280,6
120 C=255:S=199:W=50:G=0
130 Z$=CHR$(19)
140 :
150 :
160 REM  G.A.L.E.R.A.  SOFTWARE
170 :
180 REM  R A N O C C H I O
190 :
200 REM  DI GALUZZI LUCA
210 :
220 :
230 V=53248
240 POKE V+21,0
250 RESTORE
260 FOR  A=16000  TO  16383
270 READ  B : POKE A,B
280 NEXT
290 PRINT CHR$(19)"       PUNTI":RE
    M  6 SPAZI
300 Q=PEEK(V+30)
310 POKE 2040,250
320 POKE 2041,251
330 POKE 2042,252
340 POKE 2043,253
350 POKE 2044,254
360 POKE 2045,255
370 POKE V+28,255
380 POKE V,C
390 POKE V+1,100
400 POKE V+2,50
410 POKE V+3,220
420 POKE V+4,26
430 POKE V+5,220
440 POKE V+6,50
450 POKE V+7,199
460 POKE V+8,26
470 POKE V+9,199
480 POKE V+10,26
490 POKE V+11,220
500 POKE V+21,7
510 POKE V+40,2
520 POKE V+39,14
530 POKE V+38,5
540 POKE V+37,0
550 POKE V+42,2
560 REM  **** M A I N ****
570 C=255:POKE V+21,PEEK(V+21) OR
    1:POKE V+1,(INT(RND(1)*100))+5
    0
580 C=C-5:IF C<=0 THEN 570
590 POKE V,C
600 IF Z<>0 THEN 630
610 IF PEEK(203)<>64 AND Z=0 THEN
    Z=1:GOTO 630
620 GOTO 580
630 POKE V+21,PEEK(V+21) AND 249
640 POKE V+21,PEEK(V+21) OR 56
650 IF Z=1 THEN S=S-15
660 IF Z=2 THEN S=S+25
670 IF S<70 THEN W=W+10:Z=2
680 IF W>245 THEN 120
690 IF S>199 THEN Z=0:S=199:POKE V
    +21,7
700 POKE V+2,W:POKE V+4,W-24:POKE
    V+6,W:POKE V+8,W-24:POKE V+10,
    W-24
710 POKE V+7,S:POKE V+9,S:POKE V+1
    1,S+21
720 P=PEEK(V+30) AND  8 :IF P=8
    THEN GOSUB  740
730 GOTO 580
740 POKE V+21,120:C=255:G=G+117
750 PRINTZ$;G
760 RETURN
770 REM  INSETTO
780 DATA  0, 128, 0, 2, 160, 0, 2,
     160, 0
790 DATA  2, 160, 16, 82, 160, 64,
     6, 161, 5
800 DATA  6, 161, 16, 193, 135, 25
    2, 51, 247, 119
```

```
810 DATA  15, 247, 119, 51, 247, 1
    19, 193, 135, 252
820 DATA  6, 161, 16, 6, 161, 5, 8
    2, 160, 64
830 DATA  2, 160, 16, 2, 160, 0, 2
    , 160, 0
840 DATA  0, 128, 0, 0, 0, 0, 0, 0
    , 0 ,0
850 REM  TESTA DI RANA
860 DATA  0, 0, 0, 0, 3, 192, 252,
     15, 240
870 DATA  255, 207, 252, 255, 255,
     255, 255, 255, 255
880 DATA  255, 255, 255, 255, 250,
     191, 255, 234, 191
890 DATA  255, 234, 255, 255, 255,
     252, 255, 255, 240
900 DATA  87, 255, 192, 255, 255,
    0, 255, 240, 0
910 DATA  223, 0, 0, 252, 0, 0, 25
    2, 0, 0
920 DATA  255, 0, 0, 255, 0, 0, 63
    , 0, 0,0
930 REM  CODA DI RANA
940 DATA  0, 0, 0, 0, 0, 0, 0, 0,
    15
950 DATA  0, 0, 255, 0, 3, 255, 0,
     63, 255
960 DATA  0, 255, 255, 15, 255, 25
    5, 15, 255, 255
970 DATA  63, 255, 255, 63, 95, 25
    5, 63, 247, 255
```

```
980 DATA  63, 253, 127, 7, 255, 22
    1, 61, 255, 221
990 DATA  63, 255, 221, 61, 95, 22
    1, 63, 245. 80
1000 DATA  15, 255, 252, 3, 255, 25
    2, 0, 15, 252 ,0
1010 REM  TESTA DI RANA SALTANTE
1020 DATA  0, 0, 0, 0, 3, 192, 0, 1
    5, 240
1030 DATA  15, 207, 252, 255, 255,
    255, 255, 255, 255
1040 DATA  255, 255, 171, 255, 254,
     175, 255, 250, 191
1050 DATA  255, 255, 255, 255, 127,
     252, 255, 119, 240
1060 DATA  255, 125, 240, 255, 127,
     15, 255, 124, 0
1070 DATA  255, 252, 0, 252, 252, 0
    , 252, 252, 0
1080 DATA  240, 255, 192, 192, 63,
    240, 192, 63, 240,0
1090 REM  CODA DI RANA SALTANTE
1100 DATA  0, 0, 0, 0, 0, 0, 0, 0,
    0
1110 DATA  0, 0, 0, 0, 0, 0, 0, 0,
    15
1120 DATA  0, 0, 63, 0, 0, 255, 0,
    3, 255
1130 DATA  0, 3, 255, 0, 15, 255, 0
    , 63, 255
1140 DATA  0, 63, 255, 0, 63, 255,
    0, 255, 255
1150 DATA  0, 255, 255, 3, 255, 255
    , 3, 255, 255
1160 DATA  3, 247, 255, 3, 253, 255
    , 3, 255, 127,0
1170 REM  ZAMPE DI RANA SALTANTE
1180 DATA  0, 255, 220, 0, 255, 220
    , 0, 63, 252
1190 DATA  0, 63, 252, 0, 15, 255,
    0, 15, 255
1200 DATA  0, 3, 255, 0, 3, 255, 0,
     15, 252
1210 DATA  0, 63, 240, 0, 255, 192,
     3, 255, 0
1220 DATA  3, 255, 0, 15, 252, 0, 1
    5, 240, 0
1230 DATA  15, 252, 0, 15, 255, 0,
    0, 255, 192
1240 DATA  0, 63, 240, 0, 3, 240, 0
    , 0, 240,0
```

Non tutti i leoni sono veramente Leoni.

Ecco come riconoscere un vero programma Leoni Informatica

Commodore 64 Software

Leoni informatica

Garanzia a vita
Anche dopo vent'anni un programma che rivelasse un difetto d'origine viene subito sostituito.

Assistenza telefonica
Una centralina telefonica risponde ad ogni chiamata. Leoni Informatica ti fornisce anche questo servizio assicurandoti tutte le informazioni che ti necessitano per la perfetta efficienza del tuo sistema.

Aggiornamento continuo
Il mondo si evolve e le necessità cambiano. Solo per questo anche i nostri programmi possono invecchiare. Noi te li sostituiamo aggiornati.

Quando per il tuo home-computer il negoziante ti offre un programma a basso costo, diffidane. Nella quasi totalità dei casi si tratta di una copia duplicata, che per di più può non girare bene. Le conseguenze, specialmente se si tratta della tua contabilità, sono facilmente immaginabili. Leoni Informatica, Azienda leader, fa programmi da sempre, e da sempre è sinonimo di altissima qualità. Riconoscere questi programmi è facile. La classica confezione bianca e blu è accuratamente sigillata. All'interno, allegate al floppy disk, vi sono le istruzioni in italiano e, cosa importantissima, la cartolina di garanzia. Inoltre Leoni Informatica è stata la prima in Italia ad offrire la garanzia a vita, l'assistenza ed il continuo aggiornamento dei suoi programmi. Al tuo negoziante chiedi quindi la qualità, l'assistenza e l'aggiornamento: chiedi i programmi Leoni Informatica.

Richiedi a Leoni Informatica l'elenco guida ai suoi programmi.

Leoni Informatica non ti abbandona mai

Leoni informatica S.r.l. - Sviluppo Software - Vendita Hardware
Via Valsolda, 21 - 20143 Milano - Tel. 02-8467378-8465072

Potete trovare i nostri programmi in tutta Italia nei punti vendita Buffetti e dai rivenditori autorizzati che espongono il nostro marchio.

BES Milano

GIOCHI
COMMODORE 64, 16, PLUS 4

Questo gioco di carte che noi abbiamo impropriamente chiamato Poker, ne assume in realtà tutte le regole.

Lo scopo principale, come è logico, sarà quello di perdere minor soldi e guadagnarne molti. La prima azione vede proprio il discorso venale sotto forma di puntata. Il programma automaticamente ti fornirà sullo schermo le possibili vincite in conformità dei risultati ottenuti.

Buone: coppie, doppie coppie, tris, full, poker, scala, scala reale.

Per chi fosse possessore di un Commodore 16 sarà necessario apportare le seguenti modifiche:

- elimiare la linea 900;
- la riga 1600 deve essere così modificata: 1600 P1=3072:P2=40.

```
100 REM *************************
150 REM * COMPUTER UTILIZZABILI *
200 REM *                       *
250 REM * COMMODORE 64 : SI     *
300 REM *                       *
350 REM * COMMODORE 20 : NO     *
400 REM *                       *
450 REM * COMMODORE 16 : SI     *
500 REM *                       *
550 REM * COMM.PLUS 4  : SI     *
600 REM *                       *
650 REM *************************
900 POKE 53280,6:POKE 53281,6
1000 PRINT"[CLEAR][8 DOWN][12 RIGHT
    ][RVS]GIOCO DEL POKER ":PRINT:
    PRINT
1100 PRINT"[4 DOWN]             [RVS
    ]STO' MISCHIANDO![RVOFF] [2 DO
    WN]"
1200 DIM G1(5),F7(35),F8(35),F9(35)
    ,O(10)
1300 FOR I=1 TO 10:READ O(I):NEXTI:
    DATA 200,50,30,20,10,7,5,2,1,-
    1
1400 DIM C5(6),C6(5),C7(5),C8(7):FS
    $="                   "
1500 M$=" "
1600 P1=1024:P2=40
1700 W=0
1800 GOSUB 21500
1900 PRINT"[CLEAR]";
2000 PRINT"[RVS]CARTE              PU
    NTEGGIO       VINCITA  "
2100 PRINT"SCALA REALE:       200   A
    1
2200 PRINT"SCALA COLORE   :    50   A
    1":PRINT"POKER          :    30
    A 1"
2300 PRINT"FULL           :    20   A
    1":PRINT"COLORE  :             10
    A 1"
2400 PRINT"SCALA :                7  A
    1":PRINT"TRIS  :                5
    A 1"
2500 PRINT"DOPPIA COPPIA :       2  A
    1"
2600 PRINT"COPPIA :              1  A
    1
2700 GOSUB 19800
2800 PRINT"SCOMMESSA   ?: DA L.1 A L
    .10000   =";:Z9=6:GOSUB 30100:D
    =VAL(Z$)
2900 ON INT((D+9999)/10000)+1GOTO 2
    2700,3100,22900
3000 GOTO 22900
3100 PRINT"[HOME]":FOR S=1 TO 9:Z$=
    STR$(O(S)*D):PRINTSPC(25)LEFT$
    (FS$,12-LEN(Z$))"L"Z$
3200 NEXTS
3300 IF LEN(C9$)<12 THEN GOSUB 2520
    0
3400 FOR Y=1 TO 5:GOSUB 24600:GOSUB
     8700:NEXTY
3500 GOSUB 25400:SX=S:GOSUB 17900
3600 GET Q$:IF Q$<>"" THEN 3600
3700 PRINT"[HOME][12 DOWN]PER SCART
    ARE USA I NUMERI, 1,2,3,4,5 ==
    =
3800 PRINT"[UP]POI [RVS]RETURN[RVOF
    F]=(SI CORREGGE CON IL TASTO C
    .)
3900 FOR I=1 TO 5
4000 G1(I)=1
4100 NEXTI
4200 P=P1+24*P2
4300 GOSUB 23400
4400 GET Q$
4500 IF Q$=""GOTO 4400
4600 Q=ASC(Q$)
4700 IF Q=13GOTO 5800
```

```
4800 IF Q$="C"GOTO 3900
4900 IF Q$<"1"GOTO 4400
5000 IF Q$>"5"GOTO 4400
5100 Y=VAL(Q$)
5200 G1(Y)=0
5300 P=P1+24*P2+8*(Y-1)
5400 M$="= VIA ="
5500 L=LEN(M$)
5600 GOSUB 23500
5700 GOTO 4400
5800 P=P1+11*P2:GOSUB 23400
5900 P=P1+12*P2:GOSUB 23400
6000 P=P1+13*P2:GOSUB 23400:P=P1+24
     *P2:GOSUB 23400
6100 FOR Y=1 TO 5
6200 IF G1(Y)=0 THEN GOSUB 16800
6300 NEXTY
6400 GOSUB 25400
6500 IF SX=SGOTO 6800
6600 GOSUB 17900
6700 SX=S
6800 FOR Y=1 TO 5
6900 IF G1(Y)=1GOTO 7200
7000 GOSUB 24600
7100 GOSUB 8700
7200 NEXTY
7300 GOSUB 25400
7400 IF SX=SGOTO 7800
7500 GOSUB 17900
7600 SX=S
7700 GOSUB 17900
7800 REM
7900 W=W+D*O(10-S):M$="VINCITA":IF
     O(10-S)<0 THEN M$="PERDITA"
8000 M$=M$+" DI QUESTA MANO L."+STR
     $(ABS(D*O(10-S))):P=P1+11*P2:G
     OSUB 23400
8100 M$="VINCITA":IF W<0 THEN M$="P
     ERDITA
8200 P=P1+12*P2:M$="TOTALE "+M$+" =
      L."+STR$(ABS(W)):GOSUB 23400
8300 M$="PER CONTINUARE BATTI IL TA
     STO RETURN...":P=P1+13*P2:GOSU
     B 23400
8400 GET G$:IF G$="" THEN 8400
8500 IF ASC(G$)<>13 THEN 8400
8600 GOTO 1900
8700 H=C2-7
8800 IF H<4 THEN H=1
8900 FOR I=1 TO 7
9000 C8(I)=H
9100 NEXTI
9200 ON C2GOTO 9300,9600,9500,10600
     ,10200,10500,10400,9900,10100,
     9800,10700,10700,10700
9300 C8(4)=2
9400 GOTO 10700
9500 C8(4)=2
9600 C8(2)=2:C8(6)=2
9700 GOTO 10700
9800 C8(2)=2:C8(6)=2
9900 C8(3)=3:C8(5)=3
10000 GOTO 10600
10100 C8(3)=3:C8(5)=3
10200 C8(4)=2
10300 GOTO 10600
10400 C8(3)=2
10500 C8(4)=3
10600 C8(1)=3:C8(7)=3
10700 A=P1+15*P2+8*(Y-1)
10800 C3=(ASC(C2$) AND 63)+128
10900 POKE A,C3
11000 A1=160
11100 IF C2=10 THEN A1=ASC("0")+128
11200 E=6
11300 GOSUB 13200
11400 C4=ASC(C1$)
11500 F7(5)=C4:F7(31)=C4
11600 F8(1)=C4:F8(35)=C4
11700 F9(1)=C4:F9(35)=C4
11800 FOR I=1 TO 7
11900 A=A+40
12000 POKE A,160
12100 ON C8(I)GOSUB 13700,14100,1470
      0,15300,15800,16300
12200 POKE A+6,160
12300 NEXTI
12400 A=A+39
12500 IF C2=10 THEN E=5
12600 GOSUB 13200
12700 POKE A+E+1,C3
12800 IF C2=10 THEN POKE A+7,ASC("0"
      )+128
12900 C6(Y)=C1
13000 C7(Y)=C2
13100 RETURN
13200 FOR I=1 TO E
13300 POKE A+I,A1
13400 A1=160
13500 NEXTI
13600 RETURN
13700 FOR X=1 TO 5
```

```
13800 POKE A+X,160
13900 NEXTX
14000 RETURN
14100 POKE A+1,160
14200 POKE A+2,160
14300 POKE A+3,C4
14400 POKE A+4,160
14500 POKE A+5,160
14600 RETURN
14700 POKE A+1,C4
14800 POKE A+2,160
14900 POKE A+3,160
15000 POKE A+4,160
15100 POKE A+5,C4
15200 RETURN
15300 FOR X=1 TO 5
15400 X1=5*(I-1)+X
15500 POKE A+X,F7(X1)
15600 NEXTX
15700 RETURN
15800 FOR X=1 TO 5
15900 X1=5*(I-1)+X
16000 POKE A+X,F8(X1)
16100 NEXTX
16200 RETURN
16300 FOR X=1 TO 5
16400 X1=5*(I-1)+X
16500 POKE A+X,F9(X1)
16600 NEXTX
16700 RETURN
16800 REM  AZZERA LA SITUAZIONE DI G
      IOCO
16900 A=P1+15*P2+8*(Y-1)-1
17000 FOR I=1 TO 9
17100 FOR X=1 TO 7
17200 POKE A+X,32
17300 NEXTX
17400 A=A+40
17500 NEXTI
17600 C6(Y)=15+2*Y
17700 C7(Y)=15+2*Y
17800 RETURN
17900 REM
18000 IF SX=0GOTO 18900
18100 P=P1+P2*(10-SX)
18200 X=0
18300 FOR I=0 TO 37
18400 N=PEEK(P+I)
18500 IF (N=32) AND (X=0)GOTO 18800
18600 X=1
18700 POKE P+I,(N+128) AND 255
18800 NEXTI
18900 RETURN
19000 W1=20*D
19100 RETURN
19200 W1=30*D
19300 RETURN
19400 W1=50*D
19500 RETURN
19600 W1=200*D
19700 RETURN
19800 PRINT
19900 IF W=0 THEN PRINT
20000 IF W<0 THEN PRINT"IL TOTALE DE
      LLA PERDITA E' L.";-W
20100 IF W>0 THEN PRINT"IL TOTALE DE
      LLA VINCITA E' L.";W
20200 RETURN
20300 F$=" ← ) G*5T G    G   H    H
      Y64H ◤ ◥ "
20400 RETURN
20500 F$=" )#T ◤G 5      ;     <
      6 H) Y$◤ "
20600 RETURN
20700 F$=" ←  ) ' &B◤    ■     ■   )]
      & % ◤  ◥ "
20800 RETURN
20900 F7(X1)=F1
21000 RETURN
21100 F8(X1)=F1
21200 RETURN
21300 F9(X1)=F1
21400 RETURN
21500 FOR X=1 TO 3
21600 FOR X1=1 TO 35
21700 ON XGOSUB 20300,20500,20700
21800 F1=ASC(MID$(F$,X1,1))
21900 IF F1<96GOTO 22100
22000 IF F1<192 THEN F1=F1+64
22100 IF F1<64 THEN F1=F1+64
22200 F1=(F1+128) AND 255
22300 ON XGOSUB 20900,21100,21300
22400 NEXTX1
22500 NEXTX
22600 RETURN
22700 PRINT:PRINT"SCOMMESSA NON VALI
      DA"
22800 GOTO 23000
22900 PRINT:PRINT"SCOMMESSA TROPPO A
      LTA =MASSIMO L.10.000
23000 X=100
23100 X=X-1
```

```
23200 IF X>0GOTO 23100
23300 GOTO 1900
23400 L=40
23500 A=P-1:L1=LEN(M$):FOR I=1 TO L1
      :M1$=MID$(M$,I,1):POKE A+I,ASC
      (M1$) AND 63:NEXTI
23600 M1$=MID$(M$,I,1)
23700 A=A+L1
23800 L=L-L1
23900 IF L<=0GOTO 24300
24000 FOR I=1 TO L
24100 POKE A+I,32
24200 NEXTI
24300 M$=" "
24400 RETURN
24500 K
24600 C=INT(LEN(C9$)*RND(TI))+1:IF C
      >LEN(C9$) THEN C=LEN(C9$)
24700 C=INT(LEN(C9$)*RND(TI))+1:IF C
      >LEN(C9$) THEN C=LEN(C9$)
24800 C0=ASC(MID$(C9$,C,1)):C1=INT((
      C0-1)/13)+1:C2=C0-13*(C1-1)
24900 C1$=MID$("♠♥♦♣",C1,1):C2$=MID$
      ("A234567891JQK",C2,1)
25000 T$="":IF C>1 THEN T$=LEFT$(C9$
      ,C-1)
25100 C9$=T$+MID$(C9$,C+1):RETURN
25200 PRINT"[HOME][10 DOWN][RVS]LE C
      ARTE SONO STATE MESCOLATE ! "
25300 C9$="":FOR ZZ=1 TO 52:C9$=C9$+
      CHR$(ZZ):NEXTZZ:RETURN
25400 S=0
25500 S2=0
25600 FOR I=1 TO 6
25700 C5(I)=100
25800 NEXTI
25900 FOR I=1 TO 5
26000 X=5
26100 IF C7(I)>C5(X)GOTO 26500
26200 C5(X+1)=C5(X)
26300 X=X-1
26400 IF X>0GOTO 26100
26500 C5(X+1)=C7(I)
26600 NEXTI
26700 IF C5(1)=1 THEN C5(6)=14
26800 S1=0
26900 FOR I=1 TO 4
27000 IF C6(I)=C6(I+1) THEN S1=S1+1
27100 NEXTI
27200 IF S1=4 THEN S=5
27300 S1=0:FOR I=1 TO 4:IF C5(I)+5-I
      =C5(5) THEN S1=S1+1
27400 NEXTI
27500 IF S1=4 THEN S=S+3:IF S=3 THEN
       S=4
27600 IF S1=4 THEN 28000
27700 S1=0:FOR I=2 TO 5:IF C5(I)+6-I
      =C5(6) THEN S1=S1+1
27800 NEXTI
27900 IF S1=4 THEN S=S+3:IF S=3 THEN
       S=4
28000 IF (S=8) AND (C5(5)=13) THEN S
      =S+1
28100 S1=0
28200 FOR I=1 TO 4
28300 IF C5(I)=C5(I+1) THEN S1=S1+1
28400 IF C5(I)=C5(I+2) THEN S1=S1+1
28500 NEXTI
28600 IF S1>3 THEN S1=S1+2
28700 IF S1>S THEN S=S1
28800 RETURN
28900 FOR I=1 TO 5
29000 INPUT "INTRODUCI IL NUMERO DEL
      LA CARTA";C6(I),C7(I)
29100 NEXTI
29200 GOSUB 25400
29300 PRINT
29400 PRINT"S=";S
29500 PRINT
29600 PRINT"C5","C6","C7
29700 FOR I=1 TO 5
29800 PRINTC5(I),C6(I),C7(I)
29900 NEXTI:PRINTC5(6)
30000 GOTO 28900
30100 Z$="":Z0$=" ":Z1$="▒"
30200 PRINTZ0$"[LEFT]";:FOR ZZ=1 TO
      50:GET Z2$:IF Z2$<>"" THEN ZZ=
      50:NEXT:GOTO 30400
30300 NEXT:Z2$=Z1$:Z1$=Z0$:Z0$=Z2$:G
      OTO 30200
30400 ZZ=ASC(Z2$) AND 127:IF ZZ=29 O
      R ZZ=17 OR ZZ=19 OR ZZ=34 OR Z
      Z=69 THEN 30200
30500 IF ZZ<>20 THEN 30800
30600 IF LEN(Z$)>0 THEN PRINTZ2$;:IF
       LEN(Z$)>1 THEN Z$=LEFT$(Z$,LE
      N(Z$)-1):GOTO 30200
30700 GOTO 30100
30800 IF ZZ<>13 THEN PRINTZ2$;:Z$=Z$
      +Z2$:IF LEN(Z$)<Z9 THEN 30200
30900 PRINT" ":RETURN
31000 END :REM  FINE DEL GIOCO
```

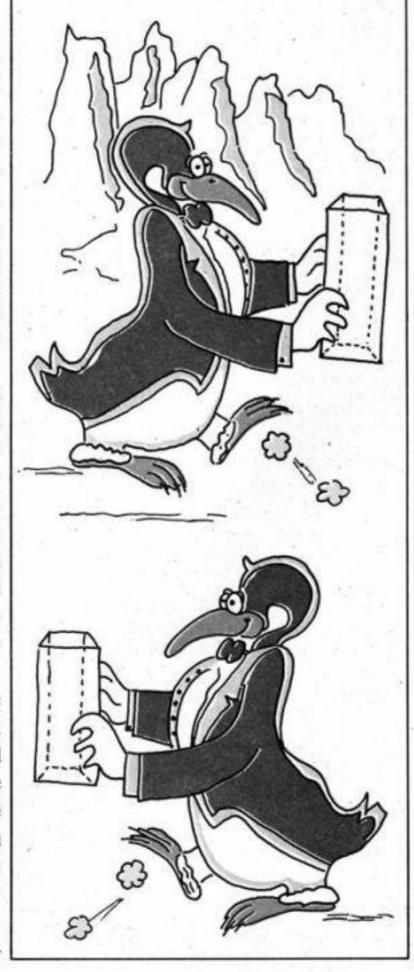
IL GIOCO DEL GHIACCIO

Trasportare ghiaccio può essere impresa difficile...

Siete stati assunti presso una ditta produttrice di ghiaccio e il vostro compito è di trasportare i blocchi dal punto di scarico fino al magazzino refrigerato. Ma, attenzione: se la giornata è molto calda il ghiaccio si scioglie velocemente e dovete essere abili e veloci. Vi occorre anche molta calma perchè rischiate di rompere il cubo. Le ganasce con cui trasportate il ghiaccio, inoltre, si usurano velocemente e la presa diminuisce lasciando scivolare il ghiaccio.

All'inizio viene chiesta la temperatura del locale. Se questa è zero o minore di zero, il ghiaccio non si scioglie e così finisce la lotta contro il tempo e il divertimento del gioco; la temperatura consigliata per iniziare è di 20 gradi centigradi. Quando iniziate a giocare le ganasce si trovano in alto a sinistra e sono indicati (a destra) il numero progressivo del blocco di ghiaccio, lo stato d'usura delle ganasce e il punteggio.

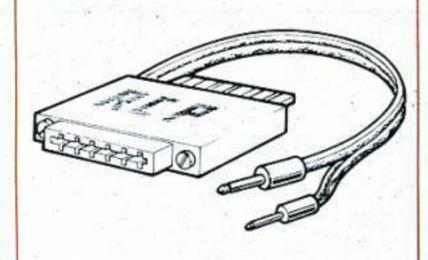
Tempestivamente dovete muovere le ganasce con i tasti W (alto), X (basso), A (sinistra), D (destra) per raggiungere il blocco posto in basso allo schermo che dovete prendere con i tasti < (allarga) e > (stringe). Attenzione a non urtarlo da una sola parte e a non stringerlo troppo: lo rompereste.

Quando è avvenuta la presa, il ghiaccio diventa di colore più scuro e, per immagazzinarlo, occorre premere la barra dello spazio. Se il ghiaccio non è stato ancora preso dalle ganasce, verrà segnalato l'errore. Più tempo impiegate per la presa, più il ghiaccio si scioglie. Dato che la presa delle ganasce è piuttosto precaria, durante il trasporto il ghiaccio potrà cadere rompendosi. Se vi fermate, si interrompe il flusso d'aria freddo creato dal movimento e il ghiaccio si scioglie molto più rapidamente. L'attrito delle ganasce sul ghiaccio diminuisce man mano si procede nel gioco a causa dell'usura. Il punteggio viene dato in base alla quantità di ghiaccio che riuscirete a salvare.

Il listato risulta volutamente "povero". Gli sprite si riducono a semplici rettangoli e la caccia al cubo, a lungo andare, non è più emozionante come all'inizio.

Compito del lettore potrebbe essere pertanto quello di "abbellire" il listato e di render difficile il gioco stesso facendo, ad esempio, apparire casualmente alcuni caratteri sullo schermo che gli sprite-ganasce devono evitare nel raggiungere l'obiettivo. Attenti a.... non scaldarvi troppo!!

Fabio Sorgato

ACCESSORI PER COMPUTER COMMODORE

DUPLICATORE PROGRAMMI

Utile accessorio per fare copie tramite un registratore commodore e un registratore normale, di nastri protetti o con caricamento turbo.

Art. CD 102 — **L. 30.000**

COPIATORE PROGRAMMI

Indispensabile accessorio per fare una copia, tramite due registratori commodore, di nastri protetti o con caricamento turbo.

Art. CD 103 — **L. 30.000**

INTERFACCIA RICEZIONE RADIO

Indispensabile per registrare su registratore commodore i programmi speciali per computer trasmessi dalle emittenti radio.

Art. CD 104 — **L. 30.000**

KIT ALLINEAMENTO TESTINE

Strumento indispensabile per la perfetta regolazione dell'AZIMUT nei registratori commodore o compatibili.

Art. CD 105 — **L. 47.000**

Via Don Pasquino Borghi, 13
42017 NOVELLARA (Reggio E.)

ALIMENTATORE

Unità di alimentazione per C 64 e VIC 20 completo di interruttore con filtro di rete e soppressore di picchicontro i disturbi elettrici. Predisposto per batterie tampone anti BLAK-OUT (Art. CD 107).

Art. CD 106 — **L. 87.000**

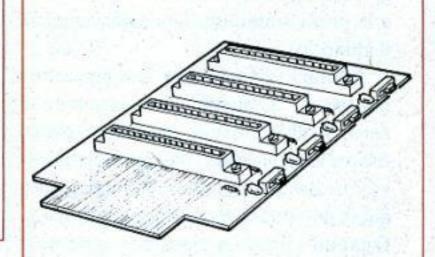
BATTERIA TAMPONE ANTI BLAK-OUT

Consente il mantenimento dei dati in memoria nei computer in assenza di corrente elettrica per circa 30 minuti (da abbinare all'alimentatore Art. CD 106). Fornibile anche senza batterie (Art. CD 117).

Art. CD 107 — **L. 85.000**

Articolo	Art.	Prezzo
Contenitore batteria tampone anti BLAK-OUT	Art. CD 117	L. 25.000
Interfaccia registratore	Art. CD 101	L. 30.000
Commutatore TV antenna/computer	Art. CD 108	L. 9.500
Tasto RESET per Vic 20 e C 64	Art. CD 109	L. 5.500
Cavo per TV con ingresso monitor	Art. CD 110	L. 18.000
Copritastiera in plexiglas	Art. CD 750	L. 18.000
Copritastiera in stoffa	Art. CD 755	L. 10.500
Borsa per trasporto C 64 e REGISTR.	Art. CD 760	L. 29.000
Vaschetta per 90 floppy D. con chiave	Art. CD 780	L. 45.000
Vaschetta per 40 floppy D. con chiave	Art. CD 770	L. 35.000
Tavola grafica SUPER SKETCH per C 64	Art. CD 790	L. 240.000
Conf. 10 dischetti ODP 1F 2D	Art. CD 700	L. 45.000
Conf. 10 dischetti KEY-DATA 1F 2D	Art. CD 710	L. 40.000
Conf. 10 dischetti COLORATI 1F 1D	Art. CD 730	L. 45.000
Conf. 10 dischetti HOBBY-FLOPPY 1F 1D	Art. CD 720	L. 29.500

TUTTI I PREZZI SONO COMPRENSIVI DI IVA
NON SI ACCETTANO ORDINI INFERIORI A L. 30.000
CONTRIBUTO FISSO SPESE DI SPEDIZIONE L. 5.000

BUS A QUATTRO SLOT PER VIC 20

Amplia la possibilità della porta di espansione fino a 4 ingressi selezionabili di cui uno indirizzabile nell'area di memoria ROM, completo di tasto RESET.

Art. CD 100 — **L. 55.000**

BUONO DI ORDINAZIONE

NOME - COGNOME

INDIRIZZO — N.

C.A.P. — CITTÀ — PROVINCIA

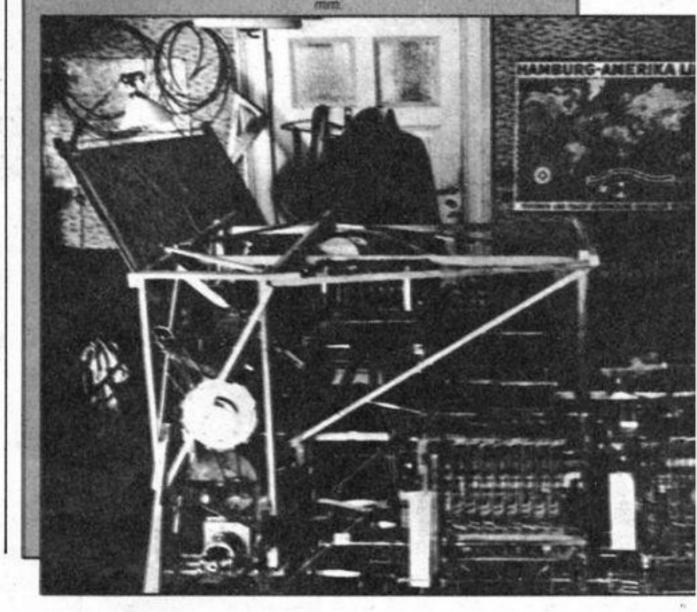
VOGLIATE INVIARMI IN CONTRASSEGNO

N.	Art.	L.
N.	Art.	L.
N.	Art.	L.
SPESE SPEDIZIONE		L. 5.000
PAGHERÒ AL POSTINO		L.

COMPUTER SERVICE VIA A. MANZONI, 49 - 42017 NOVELLARA (RE) - TEL. (0522) 661647

```
 10 REM  A COOL WORK
 20 REM  MINIGAME
 30 REM  PER COMMODORE 64
 40 REM  BY FABIO SORGATO
 41 :
 50 PRINT CHR$(147)"SPOSTAMENTI: W
    -X-A-D"
 52 PRINT"APRE CHIUDE: < - >"
 53 PRINT"PREMI UN TASTO":POKE 198
    ,0
 54 GET  A$: IF  A$="" THEN  54
100 PRINT CHR$(147)"TEMPERATURA? 2
    0"
110 PRINT TAB(11)CHR$(145);:INPUT
     TM:IF TM<0 THEN TM=0
120 PRINTCHR$(147)
130 POKE  53248+21,7:REM  ABILITA
    GLI SPRITES
140 POKE  2040,13:REM  SELEZIONA S
    PRITE 0 AL BLOCCO 13
150 POKE  2041,13:REM  SELEZIONA S
    PRITE 1 AL BLOCCO 13
160 POKE  2042,14:REM  SELEZIONA S
    PRITE 2 AL BLOCCO 14
170 POKE  53248+39,0:REM  COLORA S
    PRITE 0
180 POKE  53248+40,0:REM  COLORA S
    PRITE 1
190 POKE  53248+41,3:REM  COLORA S
    PRITE 2
200 POKE 53248+23,3:REM  ESPANDE Y
     SPRITE 2
210 SC=30:BL=1:DI=.0025*TM
220 A=50:X=A:Y=0:D=1:R=896:O=0:Q=0
230 FOR K=832 TO 832+128:POKE K,25
    5:NEXT:REM  RIEMPIE I BLOCCHI
    13 & 14 DEGLI SPRITES
240 POKE 53253,229:POKE 53252,INT(
    (RND(1)*150+50)/D)*D
250 PRINT CHR$(147)  TAB(18)"PUNTI
    :";PU
260 PRINT TAB(18)"ATTRITO GANASCE:
    ";SC:PRINT TAB(18)"BLOCCO:";BL
270 POKE 53248,X-A+24:POKE 53249,Y
    +50
280 POKE 53250,X+A+24:POKE 53251,Y
    +50
290 :
300 REM  ** MUOVE GANASCE **
310 :
320 GOSUB 480:C=PEEK(197):IF C=64
    THEN 320
330 IF C=60 THEN 500
340 IF  C=10  AND  X>D-1+A  THEN X
    =X-D:GOTO 430
350 IF  C=18  AND  X<230-A-D  THEN
     X=X+D:GOTO 430
360 IF  C=9   AND  Y>D-1  THEN Y=Y
    -D:GOTO 430
370 IF  C=23  AND  Y<158-D  THEN Y
    =Y+D:GOTO 430
380 :
390 REM  ** APRE/CHIUDE GANASCE **
400 :
410 IF  C=44  AND  A<49+D  AND  X>
    A  AND  A+X+24 <255  THEN A=A+
    D:GOTO 430
420 IF  C=47  AND  A>D+11  THEN A=
    A-D
430 Q=PEEK(53248+30) AND 7:O=-(O+(
    Q<>0))*(Q<>0)
440 IF O=0 THEN POKE 53248+41,3
450 IF O<0 AND Q=7 THEN POKE 53248
    +41,14
460 IF O>-3 THEN 270
```

Le immagini di questo fascicolo

Ecco il salotto di casa Zuse. In primo piano il calcolatore elettromeccanico Z1, creazione del ventiseienne Konrad Zuse. I modelli successivi (Z2, Z3, Z4 ecc.) verranno in parte finanziati durante la seconda guerra mondiale da enti di Stato. Le istruzioni necessarie al modello Z4 vengono perforate su una vecchia pellicola cinematografica a 35 mm.

```
470 POKE 53269,0:PRINT CHR$(147)"H
    AI ROTTO IL GHIACCIO":FOR T=0
    TO 999:NEXT:GOTO 720
480 R=R+DI:POKE R,0:IF R<956 THEN
    RETURN
490 POKE 53269,0:PRINT"[CLEAR]SI E
    ' SCIOLTO COMPLETAMENTE!!":FOR
    T=0 TO 999:NEXT:GOTO 720
500 IF Q<>7 THEN PRINT"[CLEAR]PREN
    DI IL GHIACCIO !!":FOR T=0 TO
    999:NEXT:PRINT"[CLEAR]":GOTO 2
    50
510 CK=0:FOR K=Y TO 0 STEP -1
520 POKE 53249,PEEK(53249)-1
530 POKE 53251,PEEK(53251)-1
540 POKE 53253,PEEK(53253)-1
550 GOSUB 670
560 NEXTK
570 FOR K=X-A TO 0 STEP -1
580 POKE 53248,PEEK(53248)-1
590 POKE 53250,PEEK(53250)-1
600 POKE 53252,PEEK(53252)-1
610 GOSUB 670
620 NEXTK
630 FOR K=PEEK(53253) TO 0 STEP -1
    :POKE 53253,K:NEXT
640 BL=BL+1:IF BL/3=INT(BL/3) THEN
     SC=SC-5
650 PU=PU+10*INT(956-R)
660 GOTO 220
670 CK=CK+1:IF CK<SC THEN RETURN
680 CK=0:POKE 53253,PEEK(53253)+1:
    Q=((PEEK(53248+30) AND 7)=7):I
    F Q THEN RETURN
690 FOR K=PEEK(53253) TO 255:POKE
    53253,K:NEXT
700 POKE 53269,0
710 PRINT CHR$(147)"E' CADUTO E SI
     E' SBRICIOLATO !!":FOR K=0 TO
     999:NEXT
720 PRINT CHR$(147)"PUNTI";PU:PRIN
    T CHR$(17)"PREMI UN TASTO PER
    GIOCARE"
730 POKE 198,0:WAIT 198,1: RUN
```

IL KERNAL

Come preannunciato la volta scorsa, questo mese esaminiamo le routine SAVE, LOAD e accessorie.

Prima di cominciare, però è bene fissare alcuni concetti basilari.

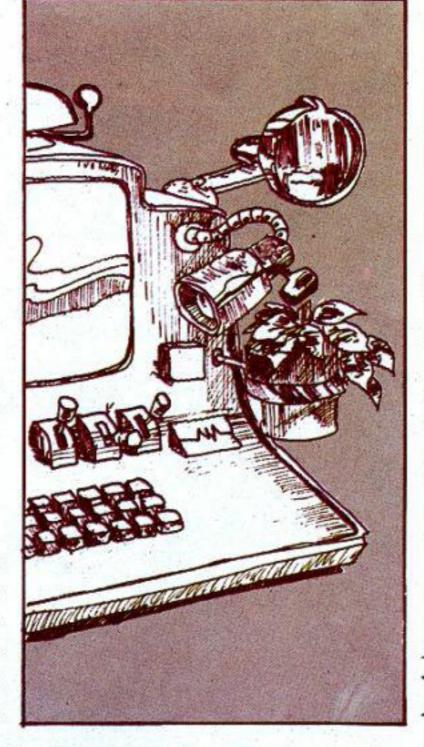
Il Basic di qualsiasi computer dispone dei comandi SAVE e LOAD che permettono, rispettivamente, di salvare su memoria esterna e caricare da memoria esterna qualsiasi tipo di programma.

Con specifico riferimento ai calcolatori Commodore, la sintassi dei due comandi è molto simile e varia a seconda delle situazioni che possono verificarsi.

Salvataggio & caricamento

Per prima cosa, se si desidera che il programma venga registrato con un nome identificatore (vengono accettati fino a 16 caratteri), bisogna aggiungere, subito dopo il comando, tale nome racchiuso tra doppi apici (può anche essere usata una variabile stringa che lo contenga): questa procedura è obbligatoria per memorizzazioni su disco.

Dopo, bisogna specificare con quale dispositivo si intende colloquiare inserendo l'appropriato numero di periferica (1 per registratore a cassette, e generalmente 8 per disk drive) precedure da una virgola separatrice (solo per il registratore, tale numero può essere omesso)

Infine, si può inserire un terzo parametro (un numero preceduto da virgola separatrice noto come indirizzo secondario) con significato diverso a seconda del tipo di memoria esterna selezionata e del tipo di Per il SAVE su nastro, questo terzo elemento può assumere il valore 1, 2 o 3.

Usando il numero 1 si decide di memorizzare il programma in modo tale che, al momento del richiamo in memoria (quando si usa il LOAD), esso venga caricato esattamente a partire dalla stessa locazione in cui si trovava al momento del salvataggio (non viene effettuata la cosiddetta rilocazione e si rivela particolarmente utile con i programmi in linguaggio macchina). Col 2 si decide di memorizzare normalmente il programma con un segnalatore finale di fine nastro mentre col 3 si hanno i due effetti precedenti combinati, cioè indirizzo di caricamento invariato e segnalatore di fine nastro.

Per il SAVE su disco, invece, è possibile inserire solamente l'indirizzo secondario 1 con lo stesso effetto visto prima per il registratore.

Con il LOAD, sia su nastro che su disco, l'indirizzo secondario può valere soltanto 1 e serve per non rilocare il programma (che sarà caricato dalla locazione in cui partiva originariamente) indipendentemente dall'indirizzo secondario usato all'atto della memorizzazione.

Come notato, il termine "rilocazione" è stato citato molto spesso in queste prime righe. D'obbligo, quindi, una spiegazione.

Normalmente, con il LOAD senza indirizzo secondario il caricamento avviene dalla locazione 2049 in poi, e ciò va bene per qualsiasi programma basic. Per quelli in linguaggio macchina, invece, questo causa sempre dei malfunzionamenti.

Sappiamo, infatti, che per far partire questi ultimi bisogna dare il comando "SYS indirizzo", dove indirizzo è la prima locazione di memoria utilizzata per la memorizzazione del programma in oggetto (quasi sempre diversa dalla 2049).

Ora, se noi carichiamo tale programma senza usare come indirizzo secondario 1, esso sarà memorizzato a partire dalla locazione 2049, e quando daremo la

SYS di partenza, non essendo presente alcun programma all'indirizzo specificato dopo tale comando avremo effetti spiacevoli variabili tra il semplice ritorno del cursore su schermo cancellato (come se si fossero premuti contemporaneamente i tasti RUN/STOP e RESTORE) e il totale blocco del sistema. Anche per un altro motivo, legato ai byte di link (non ne parliamo adesso) si consiglia sempre di caricare i programmi in linguaggio macchina con il comando:

LOAD"nome-prg",8,1 (o LOAD"nome-prg",1,1 per il registratore). Come certamente avrete già capito, quando l'indirizzo secondario viene omesso, il caricamento e il salvataggio avvengono normalmente senza le possibilità viste sopra.

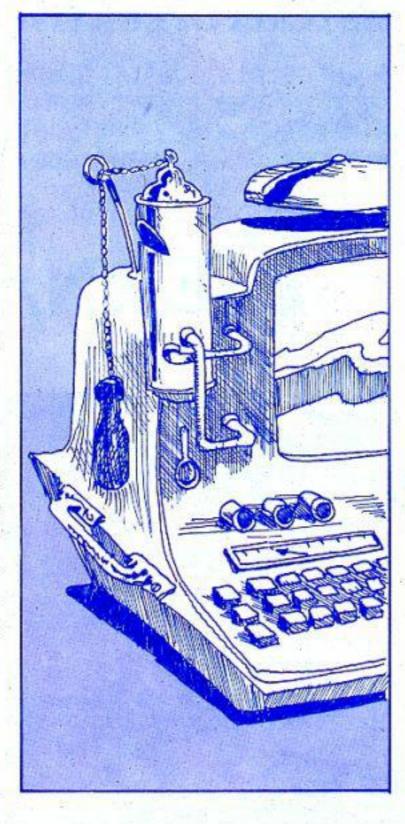
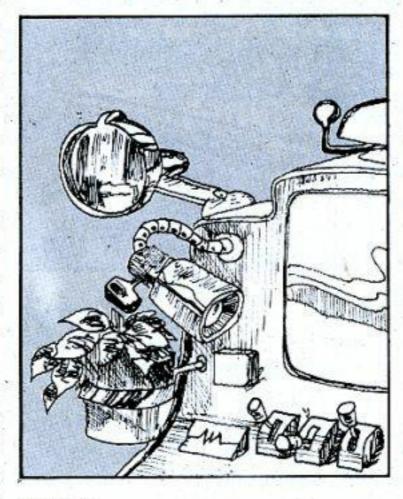
Precisato questo, cominciamo a parlare di due routine preparatorie rispetto a quelle del LOAD e del SAVE.

La routine SETLFS

E' nota col nome SETLFS, ha l'indirizzo di chiamata in 65466 ($FFBA in esadecimale) ed ha lo scopo di predisporre un file logico. Più precisamente, questa routine serve per impostare il numero logico del file, il numero della periferica e l'indirizzo secondario del comando che ha bisogno di tali elementi (nel nostro caso del LOAD o del SAVE).

Come numero logico del file si intende un qualsiasi numero, scelto arbitrariamente, che può variare da 1 a 255 (solitamente si usano numeri al di sotto di 127 in quanto gli altri possono avere speciali significati per particolari periferiche).

Per usare questa routine, basta inserire nell'accumulatore (o nella locazione 780) il numero logico del file, nel registro indice x (o nella locazione 781) il numero della periferica, nel nel registro Y (o nella locazione 782) l'indirizzo secondario e poi saltare al suo indirizzo di chiamata con un JSR (o una SYS in Basic).

Perciò, volendo predisporre un file con numero logico 2 verso l'unita a dischi e con indirizzo secondario 1, in Basic

avremo:

```
10 POKE780,2
20 POKE781,8
30 POKE782,1
40 SYS65466
```

mentre in linguaggio macchina:

```
LDA   $02
LDX   $08
LDY   $01
JSR   ffba
```

Nel caso in cui non si voglia inviare alcun indirizzo secondario, basterà inserire nel registro Y il valore 255 ($FF).

La routine SETNAM

La seconda routine preparatoria è nota come SETNAM, ha l'indirizzo di chiamata in 65469 ($FFBD) ed è usata per predisporre il nome del file. Questa routine, perciò, permette di assegnare il nome al programma che vogliamo salvare o caricare.

Per il suo uso, bisognerà impostare l'accumulatore con la lunghezza del nome del file, il registro Y con quello alto della locazione di memoria dove comincia la memorizzazione di tale nome (si rimanda alla seconda puntata dell'Assembler per tutti a pagina 43 e seguenti di C.C.C n. 8 per eventuali chiarimenti circa la dizione byte basso-byte alto), dopodiche saltare al suo indirizzo di chiamata.

Il nome può essere memorizzato, carattere per carattere in codice ASCII, in qualunque zona di memoria RAM libera per l'utente (logicamente diversa da quella occupata dal programma).

Per il linguaggio macchina è necessario inserire materialmente i codici ASCII dei caratteri del nome a partire da una locazione libera e poi il relativo valore per riportarlo nei registri X e Y. In Basic al contrario, si può usare un piccolo trucco.

Considerando il modo in cui vengono memorizzate le variabili stringa (anche in questo caso per eventuali chiarimenti sono obbligato a rimandarvi ad un altro articolo a pagina 8 e seguenti di C.C.C. n. 10) e tenendo presente che i byte 53 e 54 puntano sempre all'ultima stringa memorizzata, basterà calcolare i valori di tali puntatori, sottrarvi la lunghezza del nome ed inserire il risultato nelle locazioni 781 e 782.

Facciamo un esempio sia per il Basic

che per il linguaggio macchina. Volendo assegnare il nome VIC ad un programma, in assembly, dopo aver immesso in tre locazioni di memoria successive il codice ASCII delle lettere V, I e C (nell'esempio si è cominciato a partire da $02A7), effettuo:

```
LDA   $03
LDX   $a7
LDY   $02
JSR   ffba

.........
```

In Basic avrò:

```
10 N$= "VIC"
20 LI=PEEK(53)+PEEK(54)*256
30 LI=LI-LEN(N$)
40 POKE782,LI/256
50 POKE781,LI—PEEK(782)*256
60 POKE780,LEN(N$)
70 SYS65469
80 ..........
```

Data la semplicità del primo, commentiamo solamente il secondo listato.

Alla linea 10 si assegna ad una variabile stringa (nel nostro caso N$) il nome del file: in questo modo ci assicuriamo che in una qualche parte della memoria esso è stato memorizzato, carattere per carattere, in codice ASCII.

Per conoscere la locazione di inizio di tale memorizzazione, assegnamo dapprima alla variabile "LI" il valore del puntatore all'ultima stringa memorizzata (linea 20), dopodichè vi sottraiamo la lunghezza del nome (linea 30). Conosciuto così il valore di tale locazione, ne memorizziamo il byte alto in 782 (linea 40) e quello basso in 781 (linea 50). Per finire, memorizziamo in 780 la lunghezza del nome (linea 60) e chiamiamo questa routine con la SYS di linea 70.

L'ultima avvertenza è che, se non si vuole dare il nome al file, bisognerà solamente caricare l'accumulatore col valore 0 e saltare all'indirizzo di chiamata (non si può omettere questa sottoprocedura in quanto ai nostri fini, il suo uso è obbligatorio).

Le routines SAVE & LOAD

A questo punto possiamo passare alle routine LOAD e SAVE, cominciando con la seconda.

Scopo di questa, avente l'indirizzo di chiamata in 65496 ($FFD8), è quello di salvare un segmento di memoria su di un dispositivo di input-output.

Più precisamente, il segmento di memoria va dall'indirizzo presente in due locazioni della pagina 0 (vedremo dopo cosa significa) fino all'indirizzo memorizzato nei registri X e Y. Prima di usare questa routine, logicamente devono essere richiamate le sottoprocedure preparatorie SETLFS e SETNAM.

Da notare che il salvataggio senza usare il nome del file non genera errore solamente per il dispositivo 1 (registratore a cassette). Ora vediamo come delimit are il segmento di memoria da salvare.

Dopo aver precisato che per pagina zero si intendono le prime 256 locazioni di memoria (numerate da 0 a 255), aggiungiamo che l'indirizzamento indiretto è un metodo di programmazione frequentemente usato in linguaggio macchina che, per non appesantire la trattazione, si è

preferito non riportare in questa sede (gli interessati possono consultare le puntate sull'assembler di C.C.C.).

Ai nostri fini, comunque, basterà soltanto sapere che, per comunicare l'indirizzo di partenza della zona di memoria da salvare, bisogna inserire tale valore, sempre nel formato standard byte basso-byte alto, in due locazioni libere e consecutive della pagina 0 (ottime le 253-254) e poi caricare l'accumulatore con il numero della prima locazione utilizzata per la memorizzazione (nel nostro caso 253).

L'indirizzo finale, invece, dovrà essere contenuto nel registro X (il byte basso) e nel registro Y (il byte alto).

Precisiamo il tutto con due esempi.

Stabilito che la zona di memoria da salvare va da 4096 ($1000) a 8192 ($2000), per il linguaggio macchina avremo:

```
LDA #$00
STA $FD
LDA #$10
STA $FE
LDA #$FD
LDX #$01
LDY #$20
JSR #FFD8
BRK (o RTS)
mentre in Basic:
.........................
100 IN=4096:POKE254,IN/256
110 POKE253,IN-PEEK(254)*256
120 POKE780,253
130 FI=8192+1:POKE782,FI/256
140 POKE781,FI-PEEK(782)*256
150 SYS 65496
```

Commentiamo il primo listato: le prime 4 istruzioni inseriscono nelle locazioni della pagina 0 $FD ed $FE (253 e 254) il byte basso ed il byte alto del valore di inizio della zona di memoria da salvare ($1000), dopodichè si inserisce nell'accumulatore di valore $FD (253) che corrisponde alla locazione dove è stato memorizzato il byte basso di tale valore di inizio. Seguono le istruzioni che memorizzano nei registri X e Y l'indirizzo finale della zona di memoria da salvare (notare che quest'ultimo è stato aumentato di una unità), conclusi dal salto all'indirizzo di chiamata.

Per il listato basic, dove alle variabili IN e FI è stato assegnato il valore iniziale e finale della zona di memoria da salvare, valgono le stesse spiegazioni del listato assembly (è stata mantenuta, per quanto possibile, una certa correlazione tra le varie linee).

La serie di puntini che precede i due listati indica che, prima di questi due programmi, bisogna riportare quelli relativi alle routine SELFTS e SETNAM, senza dei quali la routine in oggetto si conclude in errore.

Inoltre, il fatto di aumentare di una unità il valore finale della zona da salvare risponde a precise esigenze del sistema ed è una pratica che bisogna sempre osservare in questi casi. L'utilità pratica di questa routine si ha quando si vuole salvare su memoria di massa un programma in linguaggio macchina o una zona di memoria richiamabili mediante la linea:

1 IFA=0THENA=1:LOAD"NOME PRG",8,1 (sostituire il ,8 con ,1 per il registratore).

Posta all'inizio del programma Basic che ne ha bisogno, mediante la routine LOAD del Kernal che segue oppure mediante il comando diretto:

LOAD"NOME PRG",8,1 anche in questo caso sostituire il ,8 con ,1 per il registratore).

Concludiamo questa puntata sul Kernal trattando la sottoprocedura LOAD, avente l'indirizzo di chiamata in 65493 ($FFD5), che ha per scopo di caricare qualsiasi programma da dispositivo esterno in RAM.

Questa routine, inoltre, può essere usata per operazioni di verifica tra il programma salvato su memoria di massa e quello presente in RAM (esattamente come il VERIFY del Basic).

L'accumulatore, prima del salto all'indirizzo di chiamata, viene impostato a 0 per il caricamento, e a 1 per la verifica. Anche per questa, prima del suo uso occorre chiamare le routine SETLFS e SETNAM.

Se si è usato nella SELFTS l'indirizzo 1, si ha un caricamento non rilocato (l'argomento "rilocazione" è stato ampiamente trattato nella prima parte dell'articolo), mentre con indirizzo secondario 0 il programma viene memorizzato a partire dalla locazione specificata dal valore dei registri X e Y, da impostare appropriatamente prima della chiamata.

Al termine dell'operazione, i registri X e Y conterranno l'indirizzo della più alta locazione RAM caricata (la fine del programma).

Per il suo uso, veramente molto semplice, basterà effettuare in linguaggio macchina:

```
LDA $00        JSR $ffd5
```

ed in Basic:

```
100 POKE780,0     110 SYS65493
```

Sempre dopo aver chiamato le routine SETLFS e SETNAM.

Come ho detto prima, nel caso si sia usato l'indirizzo secondario 1, bisognerà impostare, prima del JSR (o della SYS), i registri X e Y della locazione di inizio della memorizzazione.

Infine, qualora questa routine serve a caricare in RAM un programma Basic, subito dopo il JSR si dovranno inserire le istruzioni: STX $2D STY $2E

Per permettere ai puntatori di inizio variabili (e perciò di fine programma) di essere impostati correttamente.

Giovanni Verrelli

Caro lettore, dicci chi sei

0. Nome *(facoltativo)* ______
1. Indirizzo: ______ via ______
N. ______ città ______

2. Età:
- ☐ Meno di 15 anni ☐ 15-18 anni
- ☐ 19-24 anni ☐ 25-30 anni
- ☐ 31-45 anni ☐ 46-65 anni
- ☐ 31-45 anni

3. Sesso:
☐ M ☐ F

4. Occupazione: ______

5. Quale modello di computer possiedi?
- ☐ Vic 20 senza espansioni
- ☐ Vic 20 con espansione da 16K o oltre
- ☐ C 64
- ☐ C 16
- ☐ Altri modelli Commodore (specificare)
- ☐ Altre marche (specificare)
- ☐ Nessun computer.

6. Quali periferiche possiedi e/o intendi acquistare entro l'anno?

	Posseg.	**Intendo acquist.**
Joystick	☐	☐
Paddle	☐	☐
Tavoletta grafica	☐	☐
Penna ottica	☐	☐
Registratore a cassette	☐	☐
Floppy disk	☐	☐
Stampante	☐	☐
Plotter	☐	☐
Printer/plotter	☐	☐
Monitor	☐	☐

7. Come hai avuto questa copia di Commodore Computer Club?
- ☐ L'ho comprata in edicola ☐ Sono abbonato
- ☐ Me l'ha passata un amico/familiare ☐ Altro

8. Da quanto tempo leggi Commodore Computer Club?

	Regolar.	Saltuar.
Questa è la prima volta	☐	☐
Da qualche mese	☐	☐
Da oltre un anno	☐	☐
Fin dai primi mesi	☐	☐

9. Quali altre pubblicazioni leggi e apprezzi di più?

	Regolar.	Saltuar.
Applicando	☐	☐
Basic (enciclopedia)	☐	☐
Bit	☐	☐
Chip	☐	☐
Commodore	☐	☐
Commodore Computer Club	☐	☐
Computer	☐	☐
Computer & Electronics	☐	☐
Elettronica 2000	☐	☐
Elettronica & Computer	☐	☐
Enciclopedia dell'informatica	☐	☐
HC	☐	☐
Informatica Oggi	☐	☐
List	☐	☐
M & P Computer	☐	☐
Micro Top	☐	☐
MC Microcomputer	☐	☐
Nuova elettronica	☐	☐
Personal Time	☐	☐
PC Personal computer	☐	☐
Sinclair Computer	☐	☐
Sperimentare col computer	☐	☐
SuperSinc	☐	☐
SuperVic	☐	☐
Video giochi	☐	☐
Zerouno	☐	☐

10. Come valuti l'ultimo fascicolo di Commodore Computer Club?
(dai un voto da 0 a 10 per ciascuna delle seguenti voci)
- Livello professionale ______
- Varietà degli argomenti ______
- Chiarezza espositiva ______
- Approfondimento tecnico ______
- Tempestività ______
- Grafica/Impaginazione ______
- Pubblicità ______

11. Cosa ti interessa leggere di più?
(indica solo le tre voci che per te contano di più)
- ☐ Articoli didattici sul Basic
- ☐ Articoli didattici sul linguaggio macchina
- ☐ Articoli didattici sull'hardware
- ☐ Recensioni del software in commercio
- ☐ Giochi brevi in basic
- ☐ Giochi brevi in LM
- ☐ Routine e programmi di utilità
- ☐ Applicazioni utili per la casa
- ☐ Applicazioni utili per l'ufficio
- ☐ Applicazioni utili per la scuola

12. Come utilizzi il tuo computer?
(indicare solo le due applicazioni principali)
- ☐ A casa/ per giocare
- ☐ Per imparare a programmare
- ☐ Per il mio lavoro/il mio studio
- ☐ Quale strumento didattico
- ☐ Altro:

13. Quanti programmi software possiedi?
N. ____________________

Li hai comprati oppure te ne hanno regalato una copia?
- ☐ Comprati in un negozio
- ☐ Comprati in edicola
- ☐ Copiati
- ☐ Regalati

14. Quali riviste di programmi su cassetta sei solito comprare?

	Compro regolar.	Compro saltuar.
Commodore Club	☐	☐
Idea computer	☐	☐
Load'n Run	☐	☐
MondoSoft	☐	☐
Next	☐	☐
Open game	☐	☐
Peek	☐	☐
Play games	☐	☐
Play on tape	☐	☐
Poke	☐	☐
Run	☐	☐
16/48 Sinclair Computer	☐	☐
Soft	☐	☐
SuperSinc	☐	☐
SuperCommodore	☐	☐
Videoteca computer	☐	☐

15. Quante ore alla settimana dedichi al tuo computer?
☐ Quasi mai ☐ Meno di un'ora ☐ 1 - 5 ore
☐ 6 - 9 ore ☐ 10 - 19 ore ☐ 20 ore ed oltre

16. Quali altri hobby hai?
- ☐ Elettronica
- ☐ Modellismo
- ☐ Foto/cine
- ☐ Radiotrasmissioni
- ☐ Videoregistrazione
- ☐ Musica
- ☐ Collezionismo

17. Oltre te, quanti altri familiari o amici usano il tuo computer?

____

18. Oltre te, quanti altri familiari o amici leggono la tua copia di Commodore Computer Club?

____

19. A te capita leggere altre riviste di computer comprate da altri tuoi familiari o amici?
☐ si ☐ no

20. Puoi indicare i titoli principali? ____________________

____

PICCOLI ANNUNCI..

....

....

....

Caro lettore, dicci chi sei

INVIARE IN BUSTA
CHIUSA E AFFRANCANDO
SECONDO LE TARIFFE VIGENTI A

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

Due grandi software per il tuo 64

Commodore
COMPUTER CLUB

Corso completo di

Assembler 64

Impara il linguaggio macchina del 64 senza libri nè dispense

Dipingi sullo schermo del 64 senza tavoletta nè Koala, solo col tuo joystick.

In edicola

Maggio, tempo di IRPEF

Ora col tuo 64 puoi

In edicola